

OpenStack

API Complete Reference

(April 15, 2016)



OpenStack API Complete Reference

Copyright © 2010-2016 OpenStack Foundation All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Table of Contents

1. Block Storage API v2 (CURRENT)	1
1.1. API versions	4
1.1.1. List API versions	5
1.1.2. Show API v2 details	7
1.2. API extensions (extensions)	8
1.2.1. List API extensions	9
1.3. Limits (limits)	17
1.3.1. Show absolute limits	18
1.4. Volumes (volumes)	19
1.4.1. Create volume	21
1.4.2. List volumes	26
1.4.3. List volumes with details	28
1.4.4. Show volume details	32
1.4.5. Update volume	34
1.4.6. Delete volume	36
1.4.7. Create volume metadata	37
1.4.8. Show volume metadata	38
1.4.9. Update volume metadata	39
1.5. Volume type access (volumes)	40
1.5.1. Add private volume type access	41
1.5.2. Remove private volume type access	42
1.5.3. List private volume type access details	43
1.6. Volume actions (volumes, action)	43
1.6.1. Extend volume size	45
1.6.2. Reset volume statuses	46
1.6.3. Set image metadata for volume	47
1.6.4. Remove image metadata from volume	48
1.6.5. Attach volume to server	49
1.6.6. Unmanage volume	50
1.6.7. Force detach volume	51
1.6.8. Promote replicated volume	52
1.6.9. Reenable volume replication	53
1.7. Backups (backups)	53
1.7.1. Create backup	55
1.7.2. List backups	56
1.7.3. List backups with details	58
1.7.4. Show backup details	60
1.7.5. Delete backup	61
1.7.6. Restore backup	62
1.8. Backup actions (backups, action)	62
1.8.1. Force-delete backup	63
1.9. Capabilities for storage back ends (capabilities)	63
1.9.1. Show back-end capabilities	64
1.10. Quota sets extension (os-quota-sets)	65
1.10.1. Show quotas	66
1.10.2. Update quotas	68
1.10.3. Delete quotas	71
1.10.4. Get default quotas	72

1.10.5. Show quotas for user	74
1.10.6. Update quotas for user	76
1.10.7. Delete quotas for user	79
1.10.8. Show quota details for user	80
1.11. Quality of service (QoS) specifications (qos-specs)	80
1.11.1. Create QoS specification	82
1.11.2. List QoS specs	84
1.11.3. Show QoS specification details	86
1.11.4. Set keys in QoS specification	88
1.11.5. Delete QoS specification	89
1.11.6. Associate QoS specification with volume type	90
1.11.7. Disassociate QoS specification from volume type	91
1.11.8. Disassociate QoS specification from all associations	92
1.11.9. Get all associations for QoS specification	93
1.12. Volume types (types)	94
1.12.1. List volume types	95
1.12.2. Create volume type	97
1.12.3. Update volume type	99
1.12.4. Update extra specs for a volume type	101
1.12.5. Show volume type details	103
1.12.6. Delete volume type	105
1.13. Volume snapshots (snapshots)	105
1.13.1. Create snapshot	107
1.13.2. List snapshots	109
1.13.3. List snapshots with details	111
1.13.4. Show snapshot details	113
1.13.5. Update snapshot	114
1.13.6. Delete snapshot	116
1.13.7. Show snapshot metadata	117
1.13.8. Update snapshot metadata	118
1.14. Volume manage extension (os-volume-manage)	119
1.14.1. Manage existing volume	120
1.15. Volume image metadata extension (os-vol-image-meta)	121
1.15.1. Show image metadata for volume	122
1.16. Back-end storage pools	123
1.16.1. List back-end storage pools	124
1.17. Volume transfer	125
1.17.1. Create volume transfer	126
1.17.2. List volume transfers	127
1.17.3. List volume transfers, with details	128
1.17.4. Show volume transfer details	130
1.17.5. Delete volume transfer	131
1.17.6. Accept volume transfer	132
1.18. Consistency groups	132
1.18.1. List consistency groups	134
1.18.2. Create consistency group	135
1.18.3. List consistency groups with details	136
1.18.4. Create consistency group from source	138
1.18.5. Show consistency group details	139
1.18.6. Delete consistency group	140
1.18.7. Update consistency group	141

1.19. Consistency group snapshots	141
1.19.1. List consistency group snapshots	143
1.19.2. Create consistency group snapshot	144
1.19.3. List consistency group snapshots with details	145
1.19.4. Show consistency group snapshot details	146
1.19.5. Delete consistency group snapshot	147
2. Block Storage API v1 (DEPRECATED)	148
2.1. API versions	149
2.1.1. List API versions	150
2.1.2. Show API v1 details	151
2.2. Volumes	151
2.2.1. Create volume	153
2.2.2. List volumes	155
2.2.3. List volumes, with details	157
2.2.4. Show volume details	159
2.2.5. Delete volume	161
2.3. Volume types	161
2.3.1. List volume types	162
2.3.2. Create volume type	163
2.3.3. Update volume type	165
2.3.4. Update extra specs for a volume type	167
2.3.5. Show volume type details	169
2.3.6. Delete volume type	170
2.4. Snapshots	170
2.4.1. Create snapshot	171
2.4.2. List snapshots	173
2.4.3. List snapshots with details	175
2.4.4. Show snapshot details	177
2.4.5. Delete snapshot	178
2.4.6. Show snapshot metadata	179
2.4.7. Update snapshot metadata	181
2.5. Quota sets extension (os-quota-sets)	181
2.5.1. Show quotas	183
2.5.2. Update quotas	186
2.5.3. Delete quotas	189
2.5.4. Show default quotas	190
2.5.5. Show quotas for user	192
2.5.6. Update quotas for user	194
2.5.7. Delete quotas for user	197
2.5.8. Show quota details for user	198
3. Clustering API v1 (CURRENT)	200
3.1. API versions	201
3.1.1. List versions	202
3.2. Actions	202
3.2.1. List actions	203
3.2.2. Show action details	205
3.3. Build information	205
3.3.1. Show build information	206
3.4. Clusters	206
3.4.1. List clusters	207
3.4.2. Create cluster	209

3.4.3. Show cluster details	210
3.4.4. Update cluster	211
3.4.5. Delete cluster	212
3.4.6. Trigger cluster action	213
3.5. Cluster policies	213
3.5.1. List policies	214
3.5.2. Show cluster policy details	216
3.6. Events	216
3.6.1. List events	217
3.6.2. Show event details	219
3.7. Nodes	219
3.7.1. List nodes	221
3.7.2. Create node	223
3.7.3. Show node details	224
3.7.4. Update node	225
3.7.5. Delete node	226
3.7.6. Trigger node action	227
3.8. Policies	227
3.8.1. List policies	228
3.8.2. Create policy	230
3.8.3. Show policy details	232
3.8.4. Update policy	233
3.8.5. Delete policy	234
3.9. Policy types	234
3.9.1. List policy types	235
3.9.2. Show policy type details	236
3.10. Profiles	237
3.10.1. List profiles	238
3.10.2. Create profile	241
3.10.3. Show profile details	244
3.10.4. Update profile	246
3.10.5. Delete profile	248
3.11. Profile types	248
3.11.1. List profile types	249
3.11.2. Show profile type details	250
3.12. Receivers	251
3.12.1. List receivers	252
3.12.2. Create receiver	254
3.12.3. Show receiver details	255
3.12.4. Delete receiver	256
3.13. Webhooks	256
3.13.1. Trigger webhook action	257
4. Compute API (CURRENT)	258
4.1. API versions	268
4.1.1. List API versions	269
4.1.2. Show API v2.1	270
4.1.3. Show API v2	271
4.2. Servers (servers)	271
4.2.1. List servers	274
4.2.2. Create server	277
4.2.3. List details for servers	280

4.2.4. Show server details	283
4.2.5. Update server	286
4.2.6. Delete server	290
4.3. Servers multiple create (servers)	291
4.3.1. Create multiple servers	292
4.3.2. Create multiple servers with reservation ID	295
4.4. Servers - run an action (servers, action)	296
4.4.1. Add (associate) fixed IP (addFixedIp action)	299
4.4.2. Add (associate) floating IP (addFloatingIp action)	300
4.4.3. Change administrative password (changePassword action)	301
4.4.4. Confirm resized server (confirmResize action)	302
4.4.5. Create image (createImage action)	303
4.4.6. Evacuate server (evacuate action)	305
4.4.7. Force-delete server (forceDelete action)	306
4.4.8. Lock server (lock action)	307
4.4.9. Pause server (pause action)	308
4.4.10. Show console output (os-getConsoleOutput action)	309
4.4.11. Get RDP console (os-getRDPConsole action)	310
4.4.12. Get serial console (os-getSerialConsole action)	311
4.4.13. Get SPICE console (os-getSPICEConsole action)	312
4.4.14. Get VNC console (os-getVNCConsole action)	313
4.4.15. Reboot server (reboot action)	314
4.4.16. Rebuild server (rebuild action)	315
4.4.17. Remove (disassociate) fixed IP (removeFixedIp action)	318
4.4.18. Remove (disassociate) floating IP (removeFloatingIp action)	319
4.4.19. Rescue server (rescue action)	320
4.4.20. Resize server (resize action)	321
4.4.21. Restore soft-deleted instance (restore action)	322
4.4.22. Resume suspended server (resume action)	323
4.4.23. Revert resized server (revertResize action)	324
4.4.24. Shelf server (shelve action)	325
4.4.25. Shelf-offload (remove) server (shelveOffload action)	327
4.4.26. Start server (os-start action)	328
4.4.27. Stop server (os-stop action)	329
4.4.28. Unlock server (unlock action)	330
4.4.29. Unpause server (unpause action)	331
4.4.30. Unrescue server (unrescue action)	332
4.4.31. Unshelve (restore) shelved server (unshelve action)	333
4.4.32. Add security group to a server (addSecurityGroup action)	334
4.4.33. Remove security group from a server (removeSecurityGroup action)	335
4.4.34. Trigger crash dump in server (trigger_crash_dump action)	336
4.5. Servers - run an administrative action (servers, action)	336
4.5.1. Create server back up (createBackup action)	338
4.5.2. Inject network information (injectNetworkInfo action)	339
4.5.3. Migrate server (migrate action)	340
4.5.4. Live-migrate server (os-migrateLive action)	341
4.5.5. Reset networking on a server (resetNetwork action)	342
4.5.6. Reset server state (os-resetState action)	343
4.6. Servers diagnostics (servers, diagnostics)	343
4.6.1. Show server diagnostics	344

4.7. Servers IPs (servers, ips)	344
4.7.1. List IPs	346
4.7.2. Show IP details	347
4.8. Server metadata (servers, metadata)	347
4.8.1. List all metadata	349
4.8.2. Create or replace metadata items	350
4.8.3. Update metadata items	351
4.8.4. Show metadata item details	352
4.8.5. Create or update metadata item	353
4.8.6. Delete metadata item	354
4.9. Servers - list actions (servers, os-instance-actions)	354
4.9.1. List actions for server	355
4.9.2. Show server action details	356
4.10. Port interfaces (servers, os-interface)	357
4.10.1. Create interface	358
4.10.2. List port interfaces	359
4.10.3. Show port interface details	360
4.10.4. Detach interface	361
4.11. Servers password (servers, os-server-password)	361
4.11.1. Show server password	362
4.11.2. Clear admin password	363
4.12. Servers virtual interfaces (servers, os-virtual-interfaces)	363
4.12.1. List virtual interfaces	364
4.13. Servers with volume attachments (servers, os-volume_attachments)	364
4.13.1. Attach volume to server	366
4.13.2. List volume attachments	368
4.13.3. Show volume attachment details	369
4.13.4. Detach volume	370
4.14. Flavors with extended attributes (flavors)	371
4.14.1. List flavors	372
4.14.2. Create flavor	375
4.14.3. Show flavor details	376
4.14.4. Delete flavor	377
4.14.5. List flavors with details	378
4.15. Flavors access (flavors, os-flavor-access, action)	380
4.15.1. List flavor access information for given flavor	382
4.15.2. Add flavor access to tenant	383
4.15.3. Remove flavor access from tenant	384
4.16. Flavors extra-specs (flavors, os-flavor-extra-specs)	384
4.16.1. Create extra specs for a flavor	386
4.16.2. List extra specs for a flavor	387
4.16.3. Show an extra spec for a flavor	388
4.16.4. Update an extra spec for a flavor	389
4.16.5. Delete an extra spec for a flavor	390
4.17. Keypairs (keypairs)	390
4.17.1. List keypairs	391
4.17.2. Create or import keypair	392
4.17.3. Delete keypair	393
4.17.4. Show keypair details	394
4.18. Limits (limits)	394
4.18.1. Show rate and absolute limits	395

4.19. Extensions (extensions)	395
4.19.1. List extensions	397
4.19.2. Show extension details	412
4.20. Images	412
4.20.1. List images	414
4.20.2. List images details	419
4.20.3. Show image details	425
4.20.4. Delete image	427
4.21. Image metadata	427
4.21.1. Show image metadata	428
4.21.2. Create or replace image metadata	429
4.21.3. Update image metadata items	430
4.21.4. Show image metadata item details	431
4.21.5. Create or update image metadata item	432
4.21.6. Delete image metadata item	433
4.22. Guest agents (os-agents)	433
4.22.1. Create agent build	434
4.22.2. List agent builds	435
4.22.3. Delete agent build	436
4.22.4. Update agent build	437
4.23. Host aggregates (os-aggregates, action)	437
4.23.1. Create aggregate	439
4.23.2. List aggregates	440
4.23.3. Show aggregate details	441
4.23.4. Delete aggregate	442
4.23.5. Update aggregate	443
4.23.6. Add host	444
4.23.7. Create or update aggregate metadata	445
4.23.8. Remove host	446
4.24. Assisted volume snapshots (os-assisted-volume-snapshots)	446
4.24.1. Create assisted volume snapshots	448
4.24.2. Delete assisted volume snapshot	449
4.25. Availability zones (os-availability-zone)	449
4.25.1. Get availability zone information	450
4.25.2. Get detailed availability zone information	451
4.26. Bare metal nodes (os-baremetal-nodes)	452
4.26.1. Add bare metal node	453
4.26.2. List bare metal nodes	455
4.26.3. Add interface to bare metal node	456
4.26.4. Delete interface from bare metal node	457
4.26.5. Show bare metal node details	458
4.26.6. Delete bare metal node	460
4.27. Cells (os-cells, capacities)	460
4.27.1. List cells	461
4.27.2. List cells with details	463
4.27.3. Show cell data	464
4.27.4. Show cell capacities	465
4.28. Root certificates (os-certificates)	466
4.28.1. Create certificate	467
4.28.2. Show certificate details	469
4.29. Cloudpipe (os-cloudpipe)	469

4.29.1. List cloudpipes	471
4.29.2. Create cloudpipe	472
4.29.3. Update cloudpipe	473
4.30. Server consoles (servers, os-consoles, os-console-auth-token)	473
4.30.1. Create console	474
4.30.2. Lists consoles	475
4.30.3. Show console details	476
4.30.4. Delete console	477
4.30.5. Show console authentication token	478
4.31. Fixed IPs (os-fixed-ips)	478
4.31.1. Show fixed IP details	479
4.31.2. Reserve or release a fixed IP	480
4.32. Floating IP DNS records (os-floating-ip-dns)	480
4.32.1. List DNS domains	481
4.32.2. Create or update DNS domain	482
4.32.3. Delete DNS domain	483
4.32.4. Create or update DNS entry	484
4.32.5. Find unique DNS entry	485
4.32.6. Delete DNS entry	486
4.32.7. List DNS entries	487
4.33. Floating IP pools (os-floating-ip-pools)	487
4.33.1. List floating IP pools	488
4.34. Floating IPs (os-floating-ips)	488
4.34.1. List floating IP addresses	490
4.34.2. Create (allocate) floating IP address	491
4.34.3. Show floating IP address details	492
4.34.4. Delete (deallocate) floating IP address	493
4.35. Floating IPs bulk (os-floating-ips-bulk)	493
4.35.1. List floating IPs	494
4.35.2. Create floating IPs	495
4.35.3. Bulk-delete floating IPs	496
4.35.4. List floating IPs by host	497
4.36. Ping instances (os-fping)	497
4.36.1. Ping instances	498
4.36.2. Ping an instance	500
4.37. Hosts (os-hosts)	500
4.37.1. List hosts	502
4.37.2. Enable host	504
4.37.3. Show host details	505
4.37.4. Reboot host	506
4.37.5. Shut down host	507
4.37.6. Start host	508
4.38. Hypervisors (os-hypervisors)	508
4.38.1. List hypervisors	509
4.38.2. Show hypervisor statistics	510
4.38.3. Show hypervisor details	511
4.38.4. Show hypervisor uptime	513
4.39. Instance usage audit log (os-instance-usage-audit-log)	513
4.39.1. List usage audits for an instance	514
4.39.2. List usage audits before specified time	515
4.40. Migrations (os-migrations)	516

4.40.1. List migrations	517
4.41. Networks (os-networks)	518
4.41.1. Create network	519
4.41.2. List networks	521
4.41.3. Add network	523
4.41.4. Show network details	524
4.41.5. Delete network	526
4.41.6. Associate host	527
4.41.7. Disassociate host	528
4.41.8. Disassociate network	529
4.41.9. Disassociate project	530
4.42. Quota class (os-quota-class-sets)	530
4.42.1. Show quota	531
4.42.2. Update quota	532
4.43. Quota sets (os-quota-sets)	533
4.43.1. Update quotas	534
4.43.2. Revert quotas to defaults	536
4.43.3. List default quotas for tenant	537
4.43.4. List quotas with details	538
4.44. Security groups (os-security-groups)	539
4.44.1. List security groups	541
4.44.2. Create security group	542
4.44.3. Show security group details	543
4.44.4. Update security group	544
4.44.5. Delete security group	545
4.44.6. List security groups by server	546
4.45. Rules for default security group (os-security-group-default-rules)	546
4.45.1. List default security group rules	547
4.45.2. Create default security group rule	548
4.45.3. Show default security group rule details	549
4.45.4. Delete default security group rule	550
4.46. Rules for security group (os-security-group-rules)	550
4.46.1. Create security group rule	551
4.46.2. Delete security group rule	552
4.47. Create external events (os-server-external-events)	552
4.47.1. Run events	553
4.48. Server groups (os-server-groups)	553
4.48.1. List server groups	555
4.48.2. Create server group	556
4.48.3. Show server group details	557
4.48.4. Delete server group	558
4.49. Compute services (os-services)	558
4.49.1. List Compute services	559
4.49.2. Enable scheduling for a Compute service	561
4.49.3. Disable scheduling for a Compute service	562
4.49.4. Log disabled Compute service information	563
4.49.5. Delete Compute service	564
4.50. Usage reports (os-simple-tenant-usage)	564
4.50.1. List tenant usage for all tenants	565
4.50.2. Show usage details for tenant	566
4.51. Project networks (os-tenant-networks)	566

4.51.1. Create project network	568
4.51.2. List project networks	569
4.51.3. Show project network details	571
4.51.4. Delete project network	572
4.51.5. Associate host	573
4.51.6. Disassociate host	574
4.51.7. Disassociate network	575
4.51.8. Disassociate project	576
4.52. Volume extension (os-volumes, os-snapshots)	576
4.52.1. List volumes	578
4.52.2. Create volume	579
4.52.3. List volumes with details	580
4.52.4. Show volume details	581
4.52.5. Delete volume	582
4.52.6. List volume types	583
4.52.7. Show volume type details	584
4.52.8. Create snapshot	585
4.52.9. List snapshots	586
4.52.10. List snapshots with details	587
4.52.11. Show snapshot details	589
4.52.12. Delete snapshot	590
5. Database Service API v1.0 (CURRENT)	591
5.1. API versions	592
5.1.1. List versions	593
5.1.2. Show version details	594
5.2. Database instances (instances)	594
5.2.1. Create database instance	596
5.2.2. List database instances	599
5.2.3. Show database instance details	601
5.2.4. Attach configuration group	603
5.2.5. Detach configuration group	604
5.2.6. Delete database instance	605
5.2.7. Detach replica	606
5.2.8. List configuration defaults	607
5.3. Database instance actions (action)	608
5.3.1. Resize instance	609
5.3.2. Resize instance volume	610
5.3.3. Restart instance	612
5.3.4. Promote instance to replica source	613
5.3.5. Delete replication base instance	615
5.4. Databases (databases)	616
5.4.1. Create database	617
5.4.2. List instance databases	618
5.4.3. Delete database	620
5.5. Users (users)	620
5.5.1. Create user	622
5.5.2. List database instance users	625
5.5.3. Delete user	627
5.5.4. Enable root user	629
5.5.5. Show root-enabled status for database instance	631
5.5.6. Disable root user	632

5.6. Flavors (flavors)	632
5.6.1. List flavors	634
5.6.2. Show flavor details	636
5.7. Data stores (datastores)	637
5.7.1. List datastore versions	638
5.7.2. List configuration parameters	640
5.7.3. Show configuration parameter details	646
5.8. Configuration groups (configurations)	647
5.8.1. Create configuration group	648
5.8.2. List configuration groups	650
5.8.3. Delete configuration group	652
5.8.4. Patch configuration group	653
5.8.5. Show configuration group details	654
5.8.6. Update configuration group	656
5.8.7. List configuration group instances	657
6. Data Processing API v1.1 (CURRENT)	658
6.1. Plugins	660
6.1.1. List plugins	661
6.1.2. Show plugin details	663
6.1.3. Show plugin version details	664
6.2. Image registry	666
6.2.1. List images	667
6.2.2. Show image details	669
6.2.3. Register image	670
6.2.4. Unregister image	671
6.2.5. Add tags to image	672
6.2.6. Remove tags from image	674
6.3. Node group templates	675
6.3.1. List node group templates	676
6.3.2. Create node group template	678
6.3.3. Show node group template details	680
6.3.4. Delete node group template	682
6.3.5. Update node group template	683
6.4. Cluster templates	684
6.4.1. List cluster templates	685
6.4.2. Create cluster templates	688
6.4.3. Show cluster template details	690
6.4.4. Update cluster templates	693
6.4.5. Delete cluster template	695
6.5. Clusters	695
6.5.1. List available clusters	696
6.5.2. Create cluster	703
6.5.3. Create multiple clusters	706
6.5.4. Show details of a cluster	707
6.5.5. Scale cluster	710
6.5.6. Update cluster	718
6.5.7. Show progress	721
6.5.8. Delete a cluster	723
6.6. Event log	723
6.6.1. Show progress	724
6.7. Data sources	725

6.7.1. List data sources	726
6.7.2. Create data source	727
6.7.3. Show data source details	729
6.7.4. Update data source	730
6.7.5. Delete data source	731
6.8. Job binary internals	731
6.8.1. List job binary internals	732
6.8.2. Create job binary internal	733
6.8.3. Show job binary internal details	734
6.8.4. Delete job binary internal	735
6.8.5. Update job binary internal	736
6.8.6. Show job binary internal data	737
6.9. Job binaries	737
6.9.1. List job binaries	738
6.9.2. Show job binary details	739
6.9.3. Create job binary	740
6.9.4. Update job binary	741
6.9.5. Delete job binary	742
6.9.6. Show job binary data	743
6.10. Jobs	743
6.10.1. List jobs	745
6.10.2. Create job	747
6.10.3. Show job details	749
6.10.4. Remove job	750
6.10.5. Update job object	751
6.10.6. Run job	752
6.11. Job executions	753
6.11.1. List job executions	754
6.11.2. Show job execution details	757
6.11.3. Delete job execution	760
6.11.4. Update job execution	761
6.11.5. Refresh job execution status	764
6.11.6. Cancel job execution	767
6.12. Job types	769
6.12.1. List job types	770
7. Identity API v3 (CURRENT)	775
7.1. Authentication and token management (auth, tokens)	778
7.1.1. Password authentication with unscoped authorization	781
7.1.2. Password authentication with explicit unscoped authorization	783
7.1.3. Password authentication with scoped authorization	785
7.1.4. Token authentication with unscoped authorization	793
7.1.5. Token authentication with scoped authorization	795
7.1.6. Validate and show information for token	803
7.1.7. Check token	805
7.1.8. Revoke token	807
7.2. Credentials (credentials)	807
7.2.1. Create credential	808
7.2.2. List credentials	809
7.2.3. Show credential details	812
7.2.4. Update credential	813
7.2.5. Delete credential	814

7.3. Domains (domains, users, groups, roles)	814
7.3.1. List domains	815
7.3.2. Create domain	817
7.3.3. Show domain details	818
7.3.4. Update domain	819
7.3.5. Delete domain	820
7.4. Domain configuration (domains, config) (since v3.4) (EXPERIMENTAL)	820
7.4.1. Show domain configuration	822
7.4.2. Update domain configuration	823
7.4.3. Delete domain configuration	824
7.4.4. Show domain group configuration	825
7.4.5. Update domain group configuration	826
7.4.6. Delete domain group configuration	827
7.4.7. Show domain group option configuration	828
7.4.8. Update domain group option configuration	829
7.4.9. Delete domain group option configuration	830
7.5. Groups (groups, users)	830
7.5.1. Create group	832
7.5.2. List groups	833
7.5.3. Show group details	835
7.5.4. Update group	836
7.5.5. Delete group	837
7.5.6. List users in group	838
7.5.7. Add user to group	839
7.5.8. Check whether user belongs to group	840
7.5.9. Remove user from group	841
7.6. Policies (policies)	841
7.6.1. Create policy	842
7.6.2. List policies	843
7.6.3. Show policy details	845
7.6.4. Update policy	846
7.6.5. Delete policy	847
7.7. Projects (projects, users, groups, roles)	847
7.7.1. List projects	848
7.7.2. Create project	851
7.7.3. Show project details	852
7.7.4. Update project	853
7.7.5. Delete project	854
7.7.6. Enable or disable project and its subtree	855
7.7.7. Delete project subtree	857
7.8. Regions (regions) (since v3.2)	857
7.8.1. List regions	858
7.8.2. Create region	859
7.8.3. Show region details	860
7.8.4. Update region	861
7.8.5. Delete region	862
7.9. Roles (roles)	862
7.9.1. Create role	865
7.9.2. List roles	866
7.9.3. Show role details	868
7.9.4. Update role	869

7.9.5. Delete role	870
7.9.6. List roles for user on domain	871
7.9.7. Grant role to user on domain	872
7.9.8. Check whether user has role on domain	873
7.9.9. Revoke role from user on domain	874
7.9.10. List roles for group on domain	875
7.9.11. Grant role to group on domain	876
7.9.12. Check whether group has role on domain	877
7.9.13. Revoke role from group on domain	878
7.9.14. List roles for user on project	879
7.9.15. Grant role to user on project	880
7.9.16. Check whether user has role on project	881
7.9.17. Revoke role from user on project	882
7.9.18. List roles for group on project	883
7.9.19. Grant role to group on project	884
7.9.20. Check whether group has role on project	885
7.9.21. Revoke role from group on project	886
7.9.22. List role assignments	887
7.10. Service catalog and endpoints (services, endpoints)	888
7.10.1. List services	889
7.10.2. Create service	892
7.10.3. Show service details	893
7.10.4. Update service	894
7.10.5. Delete service	895
7.10.6. List endpoints	896
7.10.7. Create endpoint	903
7.10.8. Show endpoint details	904
7.10.9. Update endpoint	905
7.10.10. Delete endpoint	906
7.11. Users (users, groups, projects, roles)	906
7.11.1. List users	907
7.11.2. Create user	911
7.11.3. Show user details	912
7.11.4. Update user	913
7.11.5. Delete user	914
7.11.6. Change password for user	915
7.11.7. List groups to which a user belongs	916
7.11.8. List projects for user	917
8. Identity API v3 extensions (CURRENT)	918
8.1. OS-ENDPOINT-POLICY extension (OS-ENDPOINT-POLICY)	920
8.1.1. List policy and service endpoint associations	922
8.1.2. Show policy for endpoint	923
8.1.3. Check policy and service endpoint association	924
8.1.4. Associate policy and endpoint	925
8.1.5. Verify a policy and endpoint association	926
8.1.6. Delete a policy and endpoint association	927
8.1.7. Associate policy and service-type endpoint in a region	928
8.1.8. Verify a policy and service-type endpoint in a region association	929
8.1.9. Delete a policy and service-type endpoint in a region association	930
8.1.10. Associate policy and service-type endpoint	931
8.1.11. Verify a policy and service-type endpoint association	932

8.1.12. Delete a policy and service-type endpoint association	933
8.2. OS-INHERIT extension (OS-INHERIT)	933
8.2.1. List role assignments	935
8.2.2. List roles for user	938
8.2.3. Assign role to user	939
8.2.4. Check role for user	940
8.2.5. Revoke role from user	941
8.2.6. List roles for group	942
8.2.7. Assign role to group	943
8.2.8. Check role for group	944
8.2.9. Revoke role from group	945
8.2.10. List project roles for user in domain	946
8.2.11. Assign role to user owned by domain projects	947
8.2.12. Check project role for user in domain	948
8.2.13. Revoke project role from user in domain	949
8.2.14. List project roles for group in domain	950
8.2.15. Assign role to group in domain projects	951
8.2.16. Check project role for group in domain	952
8.2.17. Revoke project role from group in domain	953
8.3. Key Distribution Server (KDS) extension (OS-KDS)	953
8.3.1. Create key	957
8.3.2. Delete key	958
8.3.3. Generate ticket	959
8.3.4. Show group key	960
8.3.5. Create group	961
8.3.6. Delete group	962
8.4. OAuth extension (OS-OAUTH1)	962
8.4.1. Create consumer	964
8.4.2. List consumers	965
8.4.3. Show consumer details	966
8.4.4. Update consumer	967
8.4.5. Delete consumer	968
8.4.6. Show unauthorized request token	969
8.4.7. Create access token	970
8.4.8. List authorized access tokens	971
8.4.9. Get authorized access token	972
8.4.10. Revoke access token	973
8.4.11. List roles for an access token	974
8.4.12. Show role details for an access token	975
8.4.13. Get an Identity Service token	976
8.5. Trust extension (OS-TRUST)	976
8.5.1. Create trust	978
9. Identity API v2.0 (SUPPORTED)	980
9.1. API versions	980
9.1.1. List versions	981
9.1.2. Show version details	983
9.2. Extensions	983
9.2.1. List extensions	984
9.2.2. Show extension details	987
9.3. Tokens and tenants	987
9.3.1. Authenticate	988

9.3.2. List tenants	993
10. Identity Admin API v2.0 (SUPPORTED)	994
10.1. Versions	994
10.1.1. Get version details	995
10.2. Extensions	995
10.3. Tokens	995
10.3.1. Authenticate for admin API	997
10.3.2. Validate token	1002
10.3.3. Validate token (admin)	1004
10.3.4. Delete token	1005
10.3.5. List endpoints for token	1006
10.4. Users	1009
10.4.1. List users	1010
10.4.2. Create user	1013
10.4.3. Update user	1014
10.4.4. Delete user	1015
10.4.5. Show user details	1016
10.4.6. List user global roles	1017
10.5. Tenants	1017
10.5.1. List tenants	1019
10.5.2. Show tenant details, by name	1021
10.5.3. Show tenant details, by ID	1022
10.5.4. List roles for user	1023
11. Identity API v2.0 extensions (SUPPORTED)	1024
11.1. OS-KSADM admin extension	1025
11.1.1. List users	1027
11.1.2. Create user	1028
11.1.3. Update user	1029
11.1.4. Delete user	1030
11.1.5. Enable user	1031
11.1.6. List global roles for user	1032
11.1.7. Grant global role to user	1034
11.1.8. Delete global role from user	1035
11.1.9. Create tenant	1036
11.1.10. Update tenant	1037
11.1.11. Delete tenant	1038
11.1.12. List users on a tenant	1039
11.1.13. Grant roles to user on tenant	1041
11.1.14. Revoke role from user on tenant	1042
11.1.15. Create role	1043
11.1.16. Show role information by name	1044
11.1.17. Show role details, by ID	1045
11.1.18. Delete role	1046
11.1.19. List roles	1047
11.1.20. List services	1048
11.1.21. Create service	1050
11.1.22. Show service information by name	1051
11.1.23. Shows service information by ID	1052
11.1.24. Delete service	1053
11.2. OS-KSCATALOG admin extension	1053
11.2.1. List endpoints	1054

11.2.2. Get endpoint	1056
11.2.3. Create endpoint	1057
11.2.4. Delete endpoint	1058
11.2.5. List endpoint templates	1059
11.2.6. Get endpoint template	1061
11.2.7. Create endpoint template	1062
11.2.8. Update endpoint template	1063
11.2.9. Delete endpoint template.	1065
11.3. OS-KSEC2 admin extension	1065
11.3.1. List credentials	1066
11.3.2. Grant credential to user	1068
11.3.3. Get user credentials	1069
11.3.4. Update user credentials	1070
11.3.5. Delete user credentials	1071
11.3.6. List credentials by type	1072
11.4. OS-KSS3 admin extension	1073
11.4.1. List credentials	1074
11.4.2. Grant credential to user	1076
11.4.3. Get user credentials	1077
11.4.4. Update user credentials	1078
11.4.5. Revoke user credentials	1079
11.4.6. List credentials by type	1080
11.5. OS-KSVALIDATE admin extension	1081
11.5.1. Validate token	1082
11.5.2. Check token	1084
11.5.3. List endpoints for token	1085
12. Image service API v2 (CURRENT)	1087
12.1. API versions	1090
12.1.1. List API versions	1091
12.2. Images (images)	1091
12.2.1. List images	1092
12.2.2. Create image	1096
12.2.3. Show image details	1097
12.2.4. Update image	1098
12.2.5. Delete image	1101
12.2.6. Reactivate image	1102
12.2.7. Deactivate image	1103
12.3. Image data (images, file)	1103
12.3.1. Upload binary image data	1104
12.3.2. Download binary image data	1105
12.4. Image tags (images, tags)	1105
12.4.1. Add image tag	1107
12.4.2. Delete image tag	1108
12.5. Members (images, members)	1108
12.5.1. List image members	1109
12.5.2. Create image member	1111
12.5.3. Show image member details	1113
12.5.4. Update image member	1114
12.5.5. Delete image member	1115
12.6. Image schemas (schemas, image, images, member, members)	1115
12.6.1. Show images schema	1117

12.6.2. Show image schema	1122
12.6.3. Show image members schema	1127
12.6.4. Show image member schema	1129
12.7. Metadata definition resource types (since API v2.0) (metadefs, namespaces, resource_types)	1130
12.7.1. List resource types	1131
12.7.2. Create resource type association	1132
12.7.3. List resource type associations	1133
12.7.4. Remove resource type association	1134
12.8. Metadata definition namespaces (since API v2.0) (metadefs, namespaces)..	1134
12.8.1. Create namespace	1135
12.8.2. List namespaces	1137
12.8.3. Get namespaces details	1140
12.8.4. Update namespace	1143
12.8.5. Delete namespace	1144
12.9. Metadata definition properties (since API v2.0) (metadefs, namespaces, properties)	1144
12.9.1. Create property	1145
12.9.2. List property definitions	1147
12.9.3. Show property definition	1150
12.9.4. Update property definition	1152
12.9.5. Remove property definition	1154
12.10. Metadata definition objects (since API v2.0) (metadefs, namespaces, objects)	1154
12.10.1. Create object	1155
12.10.2. List object definitions	1157
12.10.3. Show object definition	1161
12.10.4. Update object definition	1163
12.10.5. Delete property definition	1165
12.11. Metadata definition tags (since API v2.0) (metadefs, namespaces, tags)..	1165
12.11.1. Create tags	1166
12.11.2. List tags	1167
12.11.3. Delete all tag definitions	1169
12.11.4. Add tag definition	1170
12.11.5. Get tag definition	1171
12.11.6. Update tag definition	1172
12.11.7. Delete tag definition	1173
12.12. Metadata definition schemas (schemas, metadefs, namespace, namespaces, object, objects, property, properties, tag, tags, resource_type, resource_types)	1173
12.12.1. Show metadata definition namespace schema	1175
12.12.2. Show metadata definition namespaces schema	1179
12.12.3. Show metadata definition object schema	1184
12.12.4. Show metadata definition objects schema	1187
12.12.5. Show metadata definition property schema	1191
12.12.6. Show metadata definition properties schema	1194
12.12.7. Show metadata definition tag schema	1197
12.12.8. Show metadata definition tags schema	1198
12.12.9. Show metadata definition namespace resource type association schema	1200

12.12.10. Show metadata definition namespace resource type associations schema	1202
12.13. Tasks (since API v2.2) (tasks)	1203
12.13.1. Create task	1205
12.13.2. List tasks	1206
12.13.3. Show task details	1207
13. Image service API v1 (SUPPORTED)	1208
13.1. API versions	1208
13.1.1. List API versions	1209
13.1.2. List API versions	1210
13.2. Images	1211
13.2.1. List images	1212
13.2.2. Create image	1214
13.2.3. List images with details	1215
13.2.4. Show image details and image binary	1217
13.2.5. Show image metadata	1218
13.2.6. Update image	1219
13.2.7. Delete image	1220
13.3. Members	1220
13.3.1. Replace member	1221
13.3.2. Add members to image	1222
13.3.3. Remove member	1223
13.4. Shared images	1223
13.4.1. List shared images	1224
14. Networking API v2.0 (CURRENT)	1225
14.1. API versions	1226
14.1.1. List API versions	1227
14.2. Networks	1227
14.2.1. List networks	1228
14.2.2. Create network	1230
14.2.3. Bulk create networks	1231
14.2.4. Show network details	1233
14.2.5. Update network	1234
14.2.6. Delete network	1235
14.3. Subnets	1235
14.3.1. List subnets	1236
14.3.2. Create subnet	1238
14.3.3. Bulk create subnet	1240
14.3.4. Show subnet details	1242
14.3.5. Update subnet	1243
14.3.6. Delete subnet	1244
14.4. Ports	1244
14.4.1. List ports	1245
14.4.2. Create port	1247
14.4.3. Bulk create ports	1248
14.4.4. Show port details	1250
14.4.5. Update port	1251
14.4.6. Delete port	1253
14.5. Service providers	1253
14.5.1. List service providers	1254
15. Networking API v2.0 extensions (CURRENT)	1255

15.1. Extensions	1260
15.1.1. List extensions	1262
15.1.2. Show extension details	1265
15.2. Networking Flavors Framework v2.0 (CURRENT) (flavor, service_profile)	1265
15.2.1. List flavors	1267
15.2.2. Create flavor	1268
15.2.3. Show flavor details	1270
15.2.4. Update flavor	1271
15.2.5. Delete flavor	1272
15.2.6. Associate flavor	1273
15.2.7. Disassociate a flavor.	1274
15.2.8. List service profiles	1275
15.2.9. Create service profile	1276
15.2.10. Show service profile details	1277
15.2.11. Update service profile	1278
15.2.12. Delete service profile	1279
15.3. FWaaS v2.0 (CURRENT) (fw, firewalls, firewall_policies, firewall_rules)	1279
15.3.1. List firewalls	1281
15.3.2. Create firewall	1282
15.3.3. Show firewall details	1283
15.3.4. Update firewall	1284
15.3.5. Delete firewall	1285
15.3.6. List firewall policies	1286
15.3.7. Create firewall policy	1287
15.3.8. Show firewall policy details	1288
15.3.9. Update firewall policy	1289
15.3.10. Delete firewall policy	1290
15.3.11. Insert rule into a firewall policy	1291
15.3.12. Remove rule from firewall policy	1292
15.3.13. List firewall rules	1293
15.3.14. Create firewall rule	1294
15.3.15. Show firewall rule details	1295
15.3.16. Update firewall rule	1296
15.3.17. Delete firewall rule	1297
15.4. LBaaS 2.0 (STABLE) (lbaas, loadbalancers, listeners, health_monitors, pools, members)	1297
15.4.1. List load balancers	1299
15.4.2. Create load balancer	1300
15.4.3. Show load balancer details	1303
15.4.4. Update load balancer	1304
15.4.5. Remove load balancer	1305
15.4.6. Show load balancer status tree	1306
15.4.7. List listeners	1308
15.4.8. Create listener	1309
15.4.9. Show listener details	1312
15.4.10. Update listener	1313
15.4.11. Remove listener	1315
15.4.12. List pools	1316
15.4.13. Create pool	1317
15.4.14. Show pool details	1319
15.4.15. Update pool	1321

15.4.16. Remove pool	1323
15.4.17. List pool members	1324
15.4.18. Add member to pool	1325
15.4.19. Show pool member details	1327
15.4.20. Update pool member	1328
15.4.21. Remove member from pool	1329
15.4.22. Create health monitor	1330
15.4.23. List health monitors	1332
15.4.24. Show health monitor details	1333
15.4.25. Update health monitor	1334
15.4.26. Remove health monitor	1336
15.5. LBaaS 1.0 (DEPRECATED) (lb, vips, health_monitors, pools, members)	1336
15.5.1. List VIPs	1338
15.5.2. Create a load balancer VIP	1339
15.5.3. Show VIP details	1340
15.5.4. Update VIP	1341
15.5.5. Delete VIP	1342
15.5.6. List health monitors	1343
15.5.7. Create a load balancer health monitor	1344
15.5.8. Show health monitor details	1345
15.5.9. Update health monitor	1346
15.5.10. Delete health monitor	1348
15.5.11. List pools	1349
15.5.12. Create a load balancer pool	1350
15.5.13. Show pool details	1351
15.5.14. Update pool	1352
15.5.15. Delete pool	1354
15.5.16. Associate health monitor with pool	1355
15.5.17. Disassociate health monitor from pool	1356
15.5.18. List members	1357
15.5.19. Create a load balancer member	1358
15.5.20. Show member details	1359
15.5.21. Update member	1360
15.5.22. Delete member	1361
15.6. Metering labels and rules (metering-labels, metering-label-rules)	1361
15.6.1. List metering labels	1362
15.6.2. Create metering label	1363
15.6.3. Show metering label details	1364
15.6.4. Delete metering label	1365
15.6.5. List metering label rules	1366
15.6.6. Create metering label rule	1367
15.6.7. Show metering label rule details	1368
15.6.8. Delete metering label rule	1370
15.7. Networks provider extended attributes (networks)	1370
15.7.1. List networks	1372
15.7.2. Create network	1373
15.7.3. Show network details	1374
15.7.4. Update network	1375
15.7.5. Delete network	1376
15.8. Networks multiple provider extension (networks)	1376
15.8.1. List networks	1377

15.8.2. Create network with multiple segment mappings	1379
15.8.3. Show details for a network with multiple segments	1381
15.9. VLAN transparency extension (networks)	1381
15.9.1. List networks with VLAN transparency attribute	1383
15.9.2. Create VLAN-transparent network	1384
15.9.3. Show VLAN-transparent network details	1385
15.10. Ports binding extended attributes (ports)	1385
15.10.1. List ports	1386
15.10.2. Create port	1388
15.10.3. Show port details	1389
15.10.4. Update port	1390
15.10.5. Delete port	1392
15.11. QoS policies (qos)	1392
15.11.1. List QoS policies	1393
15.11.2. Create QoS policy	1394
15.11.3. Show QoS policy details	1395
15.11.4. Update QoS policy	1396
15.11.5. Delete QoS policy	1397
15.11.6. List bandwidth limit rules for QoS policy	1398
15.11.7. Create bandwidth limit rule	1399
15.11.8. Show bandwidth limit rule details	1400
15.11.9. Update bandwidth limit rule	1401
15.11.10. Delete bandwidth limit rule	1402
15.12. Quotas extension (quotas)	1402
15.12.1. List quotas for tenants with non-default quota values	1403
15.12.2. List quotas for a tenant	1404
15.12.3. Update quota for a tenant	1405
15.12.4. Reset quota for a tenant	1406
15.13. Extra routes (routers)	1406
15.13.1. Update extra routes	1407
15.14. Layer-3 networking (routers, floatingips)	1408
15.14.1. List routers	1409
15.14.2. Create router	1411
15.14.3. Show router details	1413
15.14.4. Update router	1414
15.14.5. Delete router	1416
15.14.6. Add interface to router	1417
15.14.7. Delete interface from router	1419
15.14.8. List floating IPs	1421
15.14.9. Create floating IP	1422
15.14.10. Show floating IP details	1424
15.14.11. Update floating IP	1425
15.14.12. Delete floating IP	1427
15.15. Security groups (security-groups)	1427
15.15.1. List security groups	1428
15.15.2. Create security group	1430
15.15.3. Show security group	1432
15.15.4. Update security group	1434
15.15.5. Delete security group	1435
15.16. Security group rules (security-group-rules)	1435
15.16.1. List security group rules	1436

15.16.2. Create security group rule	1438
15.16.3. Show security group rule	1439
15.16.4. Delete security group rule	1440
15.17. Subnet pools extension (subnetpools)	1440
15.17.1. List subnet pools	1441
15.17.2. Create subnet pool	1443
15.17.3. Show subnet pool	1444
15.17.4. Update subnet pool	1445
15.17.5. Delete subnet pool	1447
15.18. Tag extension (tags)	1447
15.18.1. Replace all tags	1448
15.18.2. Remove all tags	1449
15.18.3. Confirm a tag	1450
15.18.4. Add a tag	1451
15.18.5. Remove a tag	1452
15.19. VPNaaS 2.0 (CURRENT) (vpn, vpnservices, ikepolicies, ipsecpolicies, end- point-groups, ipsec-site-connections)	1452
15.19.1. List VPN services	1454
15.19.2. Create VPN service	1455
15.19.3. Show VPN service details	1456
15.19.4. Update VPN service	1457
15.19.5. Remove VPN service	1458
15.19.6. List IKE policies	1459
15.19.7. Create IKE policy	1460
15.19.8. Show IKE policy details	1461
15.19.9. Update IKE policy	1462
15.19.10. Remove IKE policy	1463
15.19.11. List IPSec policies	1464
15.19.12. Create IPSec policy	1465
15.19.13. Show IPSec policy	1466
15.19.14. Update IPSec policy	1467
15.19.15. Remove IPSec policy	1468
15.19.16. List VPN endpoint groups	1469
15.19.17. Create VPN endpoint group	1470
15.19.18. Show VPN endpoint group	1471
15.19.19. Update VPN endpoint group	1472
15.19.20. Remove VPN endpoint group	1473
15.19.21. List IPSec connections	1474
15.19.22. Create IPSec connection	1475
15.19.23. Show IPSec connection	1477
15.19.24. Update IPSec connection	1478
15.19.25. Remove IPSec connection	1480
15.20. Network IP availability and usage stats	1480
15.20.1. Network IP Availability	1481
15.20.2. Network IP Availability	1483
16. Object Storage API v1 (SUPPORTED)	1484
16.1. Discoverability	1485
16.1.1. List activated capabilities	1486
16.2. Endpoints	1486
16.2.1. List endpoints	1487
16.3. Accounts	1487

16.3.1. Show account details and list containers	1489
16.3.2. Create, update, or delete account metadata	1493
16.3.3. Show account metadata	1496
16.4. Containers	1497
16.4.1. Show container details and list objects	1498
16.4.2. Create container	1502
16.4.3. Delete container	1505
16.4.4. Create, update, or delete container metadata	1507
16.4.5. Show container metadata	1511
16.5. Objects	1512
16.5.1. Get object content and metadata	1513
16.5.2. Create or replace object	1517
16.5.3. Copy object	1521
16.5.4. Delete object	1524
16.5.5. Show object metadata	1526
16.5.6. Create or update object metadata	1528
17. Orchestration API v1 (CURRENT)	1531
17.1. General API information	1533
17.2. API versions	1533
17.2.1. List versions	1534
17.3. Stacks	1534
17.3.1. Create stack	1536
17.3.2. Adopt stack	1538
17.3.3. List stack data	1540
17.3.4. Preview stack	1543
17.3.5. Preview stack update	1547
17.3.6. Find stack	1550
17.3.7. Find stack resources	1551
17.3.8. Show stack details	1552
17.3.9. Update stack	1554
17.3.10. Delete stack	1556
17.3.11. Abandon stack	1557
17.3.12. Snapshot stack	1559
17.3.13. List snapshots	1560
17.3.14. Show snapshot	1561
17.3.15. Delete snapshot	1563
17.3.16. Restore snapshot	1564
17.3.17. List outputs	1565
17.3.18. Show output	1566
17.4. Stack actions	1566
17.4.1. Suspend stack	1567
17.4.2. Resume stack	1568
17.4.3. Cancel stack update	1569
17.4.4. Check stack resources	1570
17.5. Stack resources	1570
17.5.1. List resources	1571
17.5.2. Show resource data	1573
17.5.3. Show resource metadata	1575
17.5.4. Send a signal to a resource	1576
17.6. Stack events	1576
17.6.1. Find stack events	1577

17.6.2. List stack events	1578
17.6.3. List resource events	1581
17.6.4. Show event details	1584
17.7. Templates	1585
17.7.1. Get stack template	1586
17.7.2. List template versions	1588
17.7.3. Validate template	1590
17.7.4. Show resource template	1592
17.7.5. Show resource schema	1594
17.7.6. List resource types	1596
17.8. Build info	1597
17.8.1. Show build information	1598
17.9. Software configuration	1598
17.9.1. Create configuration	1600
17.9.2. Show configuration details	1602
17.9.3. Delete config	1604
17.9.4. List deployments	1605
17.9.5. Create deployment	1607
17.9.6. Show server configuration metadata	1608
17.9.7. Show deployment details	1612
17.9.8. Update deployment	1613
17.9.9. Delete deployment	1615
17.10. Manage service	1615
17.10.1. Show orchestration engine status	1616
18. Shared File Systems API v2 (CURRENT)	1618
18.1. API extensions	1621
18.1.1. List extensions	1622
18.2. Limits	1624
18.2.1. List share limits	1625
18.3. Shares	1625
18.3.1. Create share	1628
18.3.2. List shares	1630
18.3.3. List shares with details	1633
18.3.4. Manage share	1637
18.3.5. Show share details	1639
18.3.6. Update share	1641
18.3.7. Delete share	1643
18.4. Share export locations (since API v2.9)	1643
18.4.1. List export locations	1644
18.4.2. Show single export location	1645
18.5. Share metadata	1645
18.5.1. Show share metadata	1647
18.5.2. Update share metadata	1648
18.5.3. Set share metadata	1649
18.5.4. Unset share metadata	1650
18.6. Share actions	1650
18.6.1. Grant access	1652
18.6.2. Revoke access	1653
18.6.3. List access rules	1654
18.6.4. Reset share state	1655
18.6.5. Force-delete share	1656

18.6.6. Extend share	1657
18.6.7. Shrink share	1658
18.6.8. Unmanage share	1659
18.7. Share snapshots	1659
18.7.1. Create share snapshot	1661
18.7.2. List share snapshots	1663
18.7.3. Manage share snapshot	1665
18.7.4. List share snapshots with details	1667
18.7.5. Show share snapshot details	1669
18.7.6. Update share snapshot	1671
18.7.7. Delete share snapshot	1673
18.7.8. Unmanage share snapshot	1674
18.7.9. Reset share snapshot state	1675
18.7.10. Force-delete share snapshot	1676
18.8. Share networks	1676
18.8.1. Create share network	1678
18.8.2. List share networks	1679
18.8.3. List share networks with details	1680
18.8.4. Show share network details	1682
18.8.5. Update share network	1683
18.8.6. Delete share network	1685
18.8.7. Add security service to share network	1686
18.8.8. Remove security service from share network	1687
18.9. Security services	1688
18.9.1. Create security service	1689
18.9.2. List security services	1691
18.9.3. List security services with details	1692
18.9.4. Show security service details	1694
18.9.5. Update security service	1695
18.9.6. Delete security service	1697
18.10. Share servers	1697
18.10.1. List share servers	1699
18.10.2. Show share server details	1700
18.10.3. Delete share server	1701
18.11. Share instances (since API v2.3)	1701
18.11.1. List share instances	1702
18.11.2. Show share instance details	1704
18.11.3. Reset share instance state	1705
18.11.4. Force-delete share instance	1706
18.12. Share instance export locations (since API v2.9)	1706
18.12.1. List export locations	1707
18.12.2. Show single export location	1708
18.13. Share types	1708
18.13.1. Create share type	1711
18.13.2. List share types	1713
18.13.3. List default share types	1715
18.13.4. Delete share type	1716
18.13.5. List extra specs	1717
18.13.6. Set extra spec for share type	1718
18.13.7. Unset an extra spec	1719
18.13.8. Add share type access	1720

18.13.9. Remove share type access	1721
18.13.10. Show share type access details	1722
18.14. Back-end storage pools	1722
18.14.1. List back-end storage pools	1724
18.14.2. List back-end storage pools with details	1725
18.15. Services	1726
18.15.1. List services	1727
18.15.2. Enable service	1728
18.15.3. Disable service	1729
18.16. Availability zones	1729
18.16.1. List availability zones	1730
18.17. Manage share	1730
18.17.1. Manage share	1732
18.17.2. Unmanage share	1734
18.18. Quota sets	1734
18.18.1. Show quotas	1735
18.18.2. Update quotas	1736
18.18.3. Delete quotas	1737
18.18.4. Show default quotas	1738
19. Shared File Systems API v1 (SUPPORTED)	1739
20. Telemetry API v2 (CURRENT)	1740
20.1. Alarms	1740
20.1.1. List alarms	1742
20.1.2. Create alarm	1744
20.1.3. Show alarm details	1746
20.1.4. Update alarm	1748
20.1.5. Delete alarm	1750
20.1.6. Update alarm state	1751
20.1.7. Show alarm state	1752
20.1.8. Show alarm history	1753
20.2. Meters	1753
20.2.1. List meters	1756
20.2.2. Add samples to meter	1758
20.2.3. List samples for meter	1761
20.2.4. Show meter statistics	1763
20.3. Samples	1764
20.3.1. List samples	1765
20.3.2. Show sample details	1767
20.4. Resources	1768
20.4.1. List resources	1769
20.4.2. Show resource details	1771
20.5. Capabilities	1772
20.5.1. List capabilities	1773
20.6. Events	1777
20.6.1. List events	1779
20.6.2. Show event details	1780

1. Block Storage API v2 (CURRENT)

Manages volumes and snapshots for use with the Block Storage API, also known as cinder services.

Method	URI	Description
API versions		
GET	/	Lists information for all Block Storage API versions.
GET	/v2	Shows details for Block Storage API v2.
API extensions (extensions)		
GET	/v2/{tenant_id}/extensions	Lists Block Storage API extensions.
Limits (limits)		
GET	/v2/{tenant_id}/limits	Shows absolute limits for a tenant.
Volumes (volumes)		
POST	/v2/{tenant_id}/volumes	Creates a volume.
GET	/v2/{tenant_id}/volumes{?sort, limit, marker}	Lists summary information for all Block Storage volumes that the tenant can access.
GET	/v2/{tenant_id}/volumes/detail{?sort, limit, marker}	Lists all Block Storage volumes, with details, that the tenant can access.
GET	/v2/{tenant_id}/volumes/{volume_id}	Shows details for a volume.
PUT	/v2/{tenant_id}/volumes/{volume_id}	Updates a volume.
DELETE	/v2/{tenant_id}/volumes/{volume_id}	Deletes a volume.
POST	/v2/{tenant_id}/volumes/{volume_id}/metadata	Creates metadata for a volume.
GET	/v2/{tenant_id}/volumes/{volume_id}/metadata	Shows metadata for a volume.
PUT	/v2/{tenant_id}/volumes/{volume_id}/metadata	Updates metadata for a volume.
Volume type access (volumes)		
POST	/v2/{tenant_id}/types/{volume_type}/action	Adds private volume type access to a project.
POST	/v2/{tenant_id}/types/{volume_type}/action	Removes private volume type access from a project.
GET	/v2/{tenant_id}/types/{volume_type}/os-volume-type-access	Lists project IDs that have access to private volume type.
Volume actions (volumes, action)		
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Extends the size of a volume to a requested size, in gibibytes (GiB). Specify the os-extend action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Resets the status, attach status, and migration status for a volume. Specify the os-reset_status action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Sets the image metadata for a volume. Specify the os-set_image_metadata action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Removes image metadata, by key, from a volume. Specify the os-unset_image_metadata action in the request body and the key for the metadata key and value pair that you want to remove.

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Attaches a volume to a server. Specify the os-attach action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Removes a volume from Block Storage management without removing the back-end storage object that is associated with it. Specify the os-unmanage action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Forces a volume to detach. Specify the os-force_detach action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Promotes a replicated volume. Specify the os-promote-replica action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Re-enables replication of a volume. Specify the volume-replica-reenable action in the request body.
Backups (backups)		
POST	/v2/{tenant_id}/backups	Creates a Block Storage backup from a volume.
GET	/v2/{tenant_id}/backups{?sort_key, sort_dir, limit, marker}	Lists Block Storage backups to which the tenant has access.
GET	/v2/{tenant_id}/backups/detail{?sort_key, sort_dir, limit, marker}	Lists Block Storage backups, with details, to which the tenant has access.
GET	/v2/{tenant_id}/backups/{backup_id}	Shows details for a backup.
DELETE	/v2/{tenant_id}/backups/{backup_id}	Deletes a backup.
POST	/v2/{tenant_id}/backups/{backup_id}/restore	Restores a Block Storage backup to an existing or new Block Storage volume.
Backup actions (backups, action)		
POST	/v2/{tenant_id}/backups/{backup_id}/action	Force-deletes a backup. Specify the os-force_delete action in the request body.
Capabilities for storage back ends (capabilities)		
GET	/v2/{tenant_id}/capabilities/{hostname}	Shows capabilities for a storage back end.
Quota sets extension (os-quota-sets)		
GET	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}{?usage}	Shows quotas for a tenant.
PUT	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.
DELETE	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}	Deletes quotas for a tenant so the quotas revert to default values.
GET	/v2/{tenant_id}/os-quota-sets/default	Gets default quotas for a tenant.
GET	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a tenant and user.
PUT	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a tenant and user.
DELETE	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Deletes quotas for a user so that the quotas revert to default values.
GET	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}	Shows details for quotas for a tenant and user.
Quality of service (QoS) specifications (qos-specs)		
POST	/v2/{tenant_id}/qos-specs	Creates a QoS specification.
GET	/v2/{tenant_id}/qos-specs{?sort_key, sort_dir, limit, marker}	Lists quality of service (QoS) specifications.

Method	URI	Description
GET	/v2/{tenant_id}/qos-specs/{qos_id}	Shows details for a QoS specification.
PUT	/v2/{tenant_id}/qos-specs/{qos_id}	Sets keys in a QoS specification.
DELETE	/v2/{tenant_id}/qos-specs/{qos_id}	Deletes a QoS specification.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/associate	Associates a QoS specification with a volume type.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate	Disassociates a QoS specification from a volume type.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate_all	Disassociates a QoS specification from all associations.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/associations	Lists all associations for a QoS specification.
Volume types (types)		
GET	/v2/{tenant_id}/types{?sort_key, sort_dir, limit, marker}	Lists volume types.
POST	/v2/{tenant_id}/types	Creates a volume type.
PUT	/v2/{tenant_id}/types/{volume_type_id}	Updates a volume type.
PUT	/v2/{tenant_id}/types/{volume_type_id}	Updates the extra specifications that are assigned to a volume type.
GET	/v2/{tenant_id}/types/{volume_type_id}	Shows details for a volume type.
DELETE	/v2/{tenant_id}/types/{volume_type_id}	Deletes a volume type.
Volume snapshots (snapshots)		
POST	/v2/{tenant_id}/snapshots	Creates a volume snapshot, which is a point-in-time, complete copy of a volume. You can create a volume from a snapshot.
GET	/v2/{tenant_id}/snapshots{?sort_key, sort_dir, limit, marker}	Lists all Block Storage snapshots, with summary information, that the tenant can access.
GET	/v2/{tenant_id}/snapshots/detail	Lists all Block Storage snapshots, with details, that the tenant can access.
GET	/v2/{tenant_id}/snapshots/{snapshot_id}	Shows details for a snapshot.
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}	Updates a snapshot.
DELETE	/v2/{tenant_id}/snapshots/{snapshot_id}	Deletes a snapshot.
GET	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Shows metadata for a snapshot.
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Updates metadata for a snapshot.
Volume manage extension (os-volume-manage)		
POST	/v2/{tenant_id}/os-volume-manage	Creates a Block Storage volume by using existing storage rather than allocating new storage.
Volume image metadata extension (os-vol-image-meta)		
GET	/v2/{tenant_id}/os-vol-image-meta	Shows image metadata for a volume.
Back-end storage pools		
GET	/v2/{tenant_id}/scheduler-stats/get_pools{?detail}	Lists all back-end storage pools.
Volume transfer		
POST	/v2/{tenant_id}/os-volume-transfer	Creates a volume transfer.

Method	URI	Description
GET	/v2/{tenant_id}/os-volume-transfer	Lists volume transfers.
GET	/v2/{tenant_id}/os-volume-transfer/detail	Lists volume transfers, with details.
GET	/v2/{tenant_id}/os-volume-transfer/{transfer_id}	Shows details for a volume transfer.
DELETE	/v2/{tenant_id}/os-volume-transfer/{transfer_id}	Deletes a volume transfer.
POST	/v2/{tenant_id}/os-volume-transfer/{transfer_id}/accept	Accepts a volume transfer.
Consistency groups		
GET	/v2/{tenant_id}/consistencygroups{?sort_key,sort_dir,limit,marker}	Lists consistency groups.
POST	/v2/{tenant_id}/consistencygroups	Creates a consistency group.
GET	/v2/{tenant_id}/consistencygroups/detail{?sort_key,sort_dir,limit,marker}	Lists consistency groups with details.
POST	/v2/{tenant_id}/consistencygroups/create_from_src	Creates a consistency group from source.
GET	/v2/{tenant_id}/consistencygroups/{consistencygroup_id}	Shows details for a consistency group.
POST	/v2/{tenant_id}/consistencygroups/{consistencygroup_id}/delete	Deletes a consistency group.
PUT	/v2/{tenant_id}/consistencygroups/{consistencygroup_id}/update	Updates a consistency group.
Consistency group snapshots		
GET	/v2/{tenant_id}/cgsnapshots	Lists all consistency group snapshots.
POST	/v2/{tenant_id}/cgsnapshots	Creates a consistency group snapshot.
GET	/v2/{tenant_id}/cgsnapshots/detail	Lists all consistency group snapshots with details.
GET	/v2/{tenant_id}/cgsnapshots/{cgsnapshot_id}	Shows details for a consistency group snapshot.
DELETE	/v2/{tenant_id}/cgsnapshots/{cgsnapshot_id}	Deletes a consistency group snapshot.

1.1. API versions

Method	URI	Description
GET	/	Lists information for all Block Storage API versions.
GET	/v2	Shows details for Block Storage API v2.

1.1.1. List API versions

Method	URI	Description
GET	/	Lists information for all Block Storage API versions.

Normal response codes: 200300

1.1.1.1. Request

This operation does not accept a request body.

1.1.1.2. Response

Example 1.1. List API versions: JSON response

```
{
  "versions": [
    {
      "status": "SUPPORTED",
      "updated": "2014-06-28T12:20:21Z",
      "id": "v1.0",
      "links": [
        {
          "href": "http://23.253.248.171:8776/v1/",
          "rel": "self"
        }
      ]
    },
    {
      "status": "CURRENT",
      "updated": "2012-11-21T11:33:21Z",
      "id": "v2.0",
      "links": [
        {
          "href": "http://23.253.248.171:8776/v2/",
          "rel": "self"
        }
      ]
    }
  ]
}
```

Example 1.2. List API versions: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<choices xmlns="http://docs.openstack.org/common/api/v1.0" xmlns:atom="http://
www.w3.org/2005/Atom">
  <version status="SUPPORTED" id="v1.0">
    <media-types>
      <media-type base="application/xml" type="application/vnd.
openstack.volume+xml;version=1" />
      <media-type base="application/json" type="application/vnd.
openstack.volume+json;version=1" />
    </media-types>
    <atom:link href="http://23.253.248.171:8776/v1/.xml" rel="self" />
  </version>
```

```
<version status="CURRENT" id="v2.0">
  <media-types>
    <media-type base="application/xml" type="application/vnd.
openstack.volume+xml;version=1" />
    <media-type base="application/json" type="application/vnd.
openstack.volume+json;version=1" />
  </media-types>
  <atom:link href="http://23.253.248.171:8776/v2/.xml" rel="self" />
</version>
</choices>
```

This operation does not return a response body.

1.1.2. Show API v2 details

Method	URI	Description
GET	/v2	Shows details for Block Storage API v2.

Normal response codes: 200203

1.1.2.1. Request

This operation does not accept a request body.

1.1.2.2. Response

Example 1.3. Show API v2 details: JSON response

```
{
  "choices": [
    {
      "status": "SUPPORTED",
      "media-types": [
        {
          "base": "application/xml",
          "type": "application/vnd.openstack.volume+xml;version=1"
        },
        {
          "base": "application/json",
          "type": "application/vnd.openstack.volume+json;version=1"
        }
      ],
      "id": "v1.0",
      "links": [
        {
          "href": "http://23.253.248.171:8776/v1/v2.json",
          "rel": "self"
        }
      ]
    },
    {
      "status": "CURRENT",
      "media-types": [
        {
          "base": "application/xml",
          "type": "application/vnd.openstack.volume+xml;version=1"
        },
        {
          "base": "application/json",
          "type": "application/vnd.openstack.volume+json;version=1"
        }
      ],
      "id": "v2.0",
      "links": [
        {
          "href": "http://23.253.248.171:8776/v2/v2.json",
          "rel": "self"
        }
      ]
    }
  ]
}
```

```

    }
  ]
}

```

This table shows the body parameters for the show api v2 details response:

Name	Type	Description
location	AnyURI (Required)	Full URL to a service or server.

Example 1.4. Show API v2 details: XML response

```

<?xml version="1.0" encoding="UTF-8"?>
<choices xmlns="http://docs.openstack.org/common/api/v1.0" xmlns:atom="http://
www.w3.org/2005/Atom">
  <version status="SUPPORTED" id="v1.0">
    <media-types>
      <media-type base="application/xml" type="application/vnd.
openstack.volume+xml;version=1" />
      <media-type base="application/json" type="application/vnd.
openstack.volume+json;version=1" />
    </media-types>
    <atom:link href="http://23.253.248.171:8776/v1/v2.xml" rel="self" />
  </version>
  <version status="CURRENT" id="v2.0">
    <media-types>
      <media-type base="application/xml" type="application/vnd.
openstack.volume+xml;version=1" />
      <media-type base="application/json" type="application/vnd.
openstack.volume+json;version=1" />
    </media-types>
    <atom:link href="http://23.253.248.171:8776/v2/v2.xml" rel="self" />
  </version>
</choices>

```

This operation does not return a response body.

1.2. API extensions (extensions)

Method	URI	Description
GET	/v2/{tenant_id}/extensions	Lists Block Storage API extensions.

1.2.1. List API extensions

Method	URI	Description
GET	/v2/{tenant_id}/extensions	Lists Block Storage API extensions.

Normal response codes: 200300

1.2.1.1. Request

This table shows the URI parameters for the list api extensions request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.2.1.2. Response

Example 1.5. List API extensions: JSON response

```
{
  "extensions": [
    {
      "updated": "2013-04-18T00:00:00+00:00",
      "name": "SchedulerHints",
      "links": [],
      "namespace": "http://docs.openstack.org/block-service/ext/
scheduler-hints/api/v2",
      "alias": "OS-SCH-HNT",
      "description": "Pass arbitrary key/value pairs to the scheduler."
    },
    {
      "updated": "2011-06-29T00:00:00+00:00",
      "name": "Hosts",
      "links": [],
      "namespace": "http://docs.openstack.org/volume/ext/hosts/api/v1.
1",
      "alias": "os-hosts",
      "description": "Admin-only host administration."
    },
    {
      "updated": "2011-11-03T00:00:00+00:00",
      "name": "VolumeTenantAttribute",
      "links": [],
      "namespace": "http://docs.openstack.org/volume/ext/
volume_tenant_attribute/api/v1",
      "alias": "os-vol-tenant-attr",
      "description": "Expose the internal project_id as an attribute of
a volume."
    },
    {
      "updated": "2011-08-08T00:00:00+00:00",
      "name": "Quotas",
      "links": [],
      "namespace": "http://docs.openstack.org/volume/ext/quotas-sets/
api/v1.1",
      "alias": "os-quota-sets",
```

```

        "description": "Quota management support."
    },
    {
        "updated": "2011-08-24T00:00:00+00:00",
        "name": "TypesManage",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/types-manage/
api/v1",
        "alias": "os-types-manage",
        "description": "Types manage support."
    },
    {
        "updated": "2013-07-10T00:00:00+00:00",
        "name": "VolumeEncryptionMetadata",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/os-volume-
encryption-metadata/api/v1",
        "alias": "os-volume-encryption-metadata",
        "description": "Volume encryption metadata retrieval support."
    },
    {
        "updated": "2012-12-12T00:00:00+00:00",
        "name": "Backups",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/backups/api/
v1",
        "alias": "backups",
        "description": "Backups support."
    },
    {
        "updated": "2013-07-16T00:00:00+00:00",
        "name": "SnapshotActions",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/snapshot-
actions/api/v1.1",
        "alias": "os-snapshot-actions",
        "description": "Enable snapshot manager actions."
    },
    {
        "updated": "2012-05-31T00:00:00+00:00",
        "name": "VolumeActions",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/volume-actions/
api/v1.1",
        "alias": "os-volume-actions",
        "description": "Enable volume actions\n    "
    },
    {
        "updated": "2013-10-03T00:00:00+00:00",
        "name": "UsedLimits",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/used-limits/
api/v1.1",
        "alias": "os-used-limits",
        "description": "Provide data on limited resources that are being
used."
    },
    {
        "updated": "2012-05-31T00:00:00+00:00",
        "name": "VolumeUnmanage",

```

```

        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/volume-
unmanage/api/v1.1",
        "alias": "os-volume-unmanage",
        "description": "Enable volume unmanage operation."
    },
    {
        "updated": "2011-11-03T00:00:00+00:00",
        "name": "VolumeHostAttribute",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/
volume_host_attribute/api/v1",
        "alias": "os-vol-host-attr",
        "description": "Expose host as an attribute of a volume."
    },
    {
        "updated": "2013-07-01T00:00:00+00:00",
        "name": "VolumeTypeEncryption",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/volume-type-
encryption/api/v1",
        "alias": "encryption",
        "description": "Encryption support for volume types."
    },
    {
        "updated": "2013-06-27T00:00:00+00:00",
        "name": "AvailabilityZones",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/os-
availability-zone/api/v1",
        "alias": "os-availability-zone",
        "description": "Describe Availability Zones."
    },
    {
        "updated": "2013-08-02T00:00:00+00:00",
        "name": "Qos_specs_manage",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/qos-specs/api/
v1",
        "alias": "qos-specs",
        "description": "QoS specs support."
    },
    {
        "updated": "2011-08-24T00:00:00+00:00",
        "name": "TypesExtraSpecs",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/types-extra-
specs/api/v1",
        "alias": "os-types-extra-specs",
        "description": "Type extra specs support."
    },
    {
        "updated": "2013-08-08T00:00:00+00:00",
        "name": "VolumeMigStatusAttribute",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/
volume_mig_status_attribute/api/v1",
        "alias": "os-vol-mig-status-attr",
        "description": "Expose migration_status as an attribute of a
volume."
    }

```



```

    },
    {
      "updated": "2012-08-13T00:00:00+00:00",
      "name": "CreateVolumeExtension",
      "links": [],
      "namespace": "http://docs.openstack.org/volume/ext/image-create/
api/v1",
      "alias": "os-image-create",
      "description": "Allow creating a volume from an image in the
Create Volume v1 API."
    },
    {
      "updated": "2014-01-10T00:00:00-00:00",
      "name": "ExtendedServices",
      "links": [],
      "namespace": "http://docs.openstack.org/volume/ext/
extended_services/api/v2",
      "alias": "os-extended-services",
      "description": "Extended services support."
    },
    {
      "updated": "2012-06-19T00:00:00+00:00",
      "name": "ExtendedSnapshotAttributes",
      "links": [],
      "namespace": "http://docs.openstack.org/volume/ext/
extended_snapshot_attributes/api/v1",
      "alias": "os-extended-snapshot-attributes",
      "description": "Extended SnapshotAttributes support."
    },
    {
      "updated": "2012-12-07T00:00:00+00:00",
      "name": "VolumeImageMetadata",
      "links": [],
      "namespace": "http://docs.openstack.org/volume/ext/
volume_image_metadata/api/v1",
      "alias": "os-vol-image-meta",
      "description": "Show image metadata associated with the volume."
    },
    {
      "updated": "2012-03-12T00:00:00+00:00",
      "name": "QuotaClasses",
      "links": [],
      "namespace": "http://docs.openstack.org/volume/ext/quota-classes-
sets/api/v1.1",
      "alias": "os-quota-class-sets",
      "description": "Quota classes management support."
    },
    {
      "updated": "2013-05-29T00:00:00+00:00",
      "name": "VolumeTransfer",
      "links": [],
      "namespace": "http://docs.openstack.org/volume/ext/volume-
transfer/api/v1.1",
      "alias": "os-volume-transfer",
      "description": "Volume transfer management support."
    },
    {
      "updated": "2014-02-10T00:00:00+00:00",
      "name": "VolumeManage",
      "links": [],

```

```

        "namespace": "http://docs.openstack.org/volume/ext/os-volume-
manage/api/v1",
        "alias": "os-volume-manage",
        "description": "Allows existing backend storage to be 'managed' by
Cinder."
    },
    {
        "updated": "2012-08-25T00:00:00+00:00",
        "name": "AdminActions",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/admin-actions/
api/v1.1",
        "alias": "os-admin-actions",
        "description": "Enable admin actions."
    },
    {
        "updated": "2012-10-28T00:00:00-00:00",
        "name": "Services",
        "links": [],
        "namespace": "http://docs.openstack.org/volume/ext/services/api/
v2",
        "alias": "os-services",
        "description": "Services support."
    }
]
}

```

This table shows the body parameters for the list api extensions response:

Name	Type	Description
name	String (Required)	The name of the extension. For example, "Fox In Socks."
description	String (Required)	The extension description.
namespace	String (Required)	Link associated to the extension.
alias	String (Required)	The alias for the extension. For example, "FOXNSOX", "os-availability-zone", "os-extended-quotas", "os-share-unmanage" or "os-used-limits."
links	List (Required)	List of links related to the extension.
updated	DateTime (Required)	The date and time stamp when the extension was last updated.

Example 1.6. List API extensions: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<extensions xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/common/api/v1.0">
  <extension alias="OS-SCH-HNT" updated="2013-04-18T00:00:00+00:00"
    namespace="http://docs.openstack.org/block-service/ext/scheduler-
hints/api/v2"
    name="SchedulerHints">
    <description>Pass arbitrary key/value pairs to the
      scheduler.</description>
  </extension>
</extensions>

```

```

</extension>
<extension alias="os-hosts" updated="2011-06-29T00:00:00+00:00"
  namespace="http://docs.openstack.org/volume/ext/hosts/api/v1.1"
  name="Hosts">
  <description>Admin-only host administration.</description>
</extension>
<extension alias="os-vol-tenant-attr"
  updated="2011-11-03T00:00:00+00:00"
  namespace="http://docs.openstack.org/volume/ext/
volume_tenant_attribute/api/v1"
  name="VolumeTenantAttribute">
  <description>Expose the internal project_id as an attribute of
    a volume.</description>
</extension>
<extension alias="os-quota-sets"
  updated="2011-08-08T00:00:00+00:00"
  namespace="http://docs.openstack.org/volume/ext/quotas-sets/api/v1.1"
  name="Quotas">
  <description>Quota management support.</description>
</extension>
<extension alias="os-types-manage"
  updated="2011-08-24T00:00:00+00:00"
  namespace="http://docs.openstack.org/volume/ext/types-manage/api/v1"
  name="TypesManage">
  <description>Types manage support.</description>
</extension>
<extension alias="os-volume-encryption-metadata"
  updated="2013-07-10T00:00:00+00:00"
  namespace="http://docs.openstack.org/volume/ext/os-volume-encryption-
metadata/api/v1"
  name="VolumeEncryptionMetadata">
  <description>Volume encryption metadata retrieval
    support.</description>
</extension>
<extension alias="backups" updated="2012-12-12T00:00:00+00:00"
  namespace="http://docs.openstack.org/volume/ext/backups/api/v1"
  name="Backups">
  <description>Backups support.</description>
</extension>
<extension alias="os-snapshot-actions"
  updated="2013-07-16T00:00:00+00:00"
  namespace="http://docs.openstack.org/volume/ext/snapshot-actions/api/
v1.1"
  name="SnapshotActions">
  <description>Enable snapshot manager actions.</description>
</extension>
<extension alias="os-volume-actions"
  updated="2012-05-31T00:00:00+00:00"
  namespace="http://docs.openstack.org/volume/ext/volume-actions/api/v1.
1"
  name="VolumeActions">
  <description>Enable volume actions </description>
</extension>
<extension alias="os-used-limits"
  updated="2013-10-03T00:00:00+00:00"
  namespace="http://docs.openstack.org/volume/ext/used-limits/api/v1.1"
  name="UsedLimits">
  <description>Provide data on limited resources that are being
    used.</description>
</extension>

```

```

    <extension alias="os-volume-unmanage"
      updated="2012-05-31T00:00:00+00:00"
      namespace="http://docs.openstack.org/volume/ext/volume-unmanage/api/
v1.1"
      name="VolumeUnmanage">
        <description>Enable volume unmanage operation.</description>
      </extension>
    <extension alias="os-vol-host-attr"
      updated="2011-11-03T00:00:00+00:00"
      namespace="http://docs.openstack.org/volume/ext/volume_host_attribute/
api/v1"
      name="VolumeHostAttribute">
        <description>Expose host as an attribute of a
          volume.</description>
      </extension>
    <extension alias="encryption" updated="2013-07-01T00:00:00+00:00"
      namespace="http://docs.openstack.org/volume/ext/volume-type-
encryption/api/v1"
      name="VolumeTypeEncryption">
        <description>Encryption support for volume
          types.</description>
      </extension>
    <extension alias="os-availability-zone"
      updated="2013-06-27T00:00:00+00:00"
      namespace="http://docs.openstack.org/volume/ext/os-availability-zone/
api/v1"
      name="AvailabilityZones">
        <description>Describe Availability Zones.</description>
      </extension>
    <extension alias="qos-specs" updated="2013-08-02T00:00:00+00:00"
      namespace="http://docs.openstack.org/volume/ext/qos-specs/api/v1"
      name="Qos_specs_manage">
        <description>QoS specs support.</description>
      </extension>
    <extension alias="os-types-extra-specs"
      updated="2011-08-24T00:00:00+00:00"
      namespace="http://docs.openstack.org/volume/ext/types-extra-specs/api/
v1"
      name="TypesExtraSpecs">
        <description>Type extra specs support.</description>
      </extension>
    <extension alias="os-vol-mig-status-attr"
      updated="2013-08-08T00:00:00+00:00"
      namespace="http://docs.openstack.org/volume/ext/
volume_mig_status_attribute/api/v1"
      name="VolumeMigStatusAttribute">
        <description>Expose migration_status as an attribute of a
          volume.</description>
      </extension>
    <extension alias="os-image-create"
      updated="2012-08-13T00:00:00+00:00"
      namespace="http://docs.openstack.org/volume/ext/image-create/api/v1"
      name="CreateVolumeExtension">
        <description>Allow creating a volume from an image in the
          Create Volume v1 API.</description>
      </extension>
    <extension alias="os-extended-services"
      updated="2014-01-10T00:00:00-00:00"
      namespace="http://docs.openstack.org/volume/ext/extended_services/api/
v2"

```

```

        name="ExtendedServices">
        <description>Extended services support.</description>
    </extension>
    <extension alias="os-extended-snapshot-attributes"
        updated="2012-06-19T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/
extended_snapshot_attributes/api/v1"
        name="ExtendedSnapshotAttributes">
        <description>Extended SnapshotAttributes
            support.</description>
    </extension>
    <extension alias="os-vol-image-meta"
        updated="2012-12-07T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/volume_image_metadata/
api/v1"
        name="VolumeImageMetadata">
        <description>Show image metadata associated with the
            volume.</description>
    </extension>
    <extension alias="os-quota-class-sets"
        updated="2012-03-12T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/quota-classes-sets/
api/v1.1"
        name="QuotaClasses">
        <description>Quota classes management support.</description>
    </extension>
    <extension alias="os-volume-transfer"
        updated="2013-05-29T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/volume-transfer/api/
v1.1"
        name="VolumeTransfer">
        <description>Volume transfer management support.</description>
    </extension>
    <extension alias="os-volume-manage"
        updated="2014-02-10T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/os-volume-manage/api/
v1"
        name="VolumeManage">
        <description>Allows existing back end storage to be 'managed'
            by cinder.</description>
    </extension>
    <extension alias="os-admin-actions"
        updated="2012-08-25T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/admin-actions/api/v1.
1"
        name="AdminActions">
        <description>Enable admin actions.</description>
    </extension>
    <extension alias="os-services" updated="2012-10-28T00:00:00-00:00"
        namespace="http://docs.openstack.org/volume/ext/services/api/v2"
        name="Services">
        <description>Services support.</description>
    </extension>
</extensions>

```

This operation does not return a response body.

1.3. Limits (limits)

Shows absolute limits for a tenant.

An absolute limit value of -1 indicates that the absolute limit for the item is infinite.

Method	URI	Description
GET	/v2/{tenant_id}/limits	Shows absolute limits for a tenant.

1.3.1. Show absolute limits

Method	URI	Description
GET	/v2/{tenant_id}/limits	Shows absolute limits for a tenant.

An absolute limit value of -1 indicates that the absolute limit for the item is infinite.

Normal response codes: 200203

1.3.1.1. Request

This table shows the URI parameters for the show absolute limits request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.3.1.2. Response

Example 1.7. Show absolute limits: JSON response

```
{
  "limits": {
    "rate": [],
    "absolute": {
      "totalSnapshotsUsed": 0,
      "maxTotalBackups": 10,
      "maxTotalVolumeGigabytes": 1000,
      "maxTotalSnapshots": 10,
      "maxTotalBackupGigabytes": 1000,
      "totalBackupGigabytesUsed": 0,
      "maxTotalVolumes": 10,
      "totalVolumesUsed": 0,
      "totalBackupsUsed": 0,
      "totalGigabytesUsed": 0
    }
  }
}
```

Example 1.8. Show absolute limits: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<limits xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/common/api/v1.0">
  <rates/>
  <absolute>
    <limit name="totalSnapshotsUsed" value="0"/>
    <limit name="maxTotalBackups" value="10"/>
    <limit name="maxTotalVolumeGigabytes" value="1000"/>
    <limit name="maxTotalSnapshots" value="10"/>
    <limit name="maxTotalBackupGigabytes" value="1000"/>
    <limit name="totalBackupGigabytesUsed" value="0"/>
    <limit name="maxTotalVolumes" value="10"/>
    <limit name="totalVolumesUsed" value="0"/>
  </absolute>
</limits>
```

```
<limit name="totalBackupsUsed" value="0"/>
<limit name="totalGigabytesUsed" value="0"/>
</absolute>
</limits>
```

This operation does not return a response body.

1.4. Volumes (volumes)

A volume is a detachable block storage device similar to a USB hard drive. You can attach a volume to one instance at a time.

The `snapshot_id` and `source_volume_id` parameters specify the ID of the snapshot or volume from which this volume originates. If the volume was not created from a snapshot or source volume, these values are null.

When you create, list, update, or delete volumes, the possible status values are:

Table 1.1. Volume statuses

Status	Description
creating	The volume is being created.
available	The volume is ready to attach to an instance.
attaching	The volume is attaching to an instance.
in-use	The volume is attached to an instance.
deleting	The volume is being deleted.
error	A volume creation error occurred.
error_deleting	A volume deletion error occurred.
backing-up	The volume is being backed up.
restoring-backup	A backup is being restored to the volume.
error_restoring	A backup restoration error occurred.
error_extending	An error occurred while attempting to extend a volume.

Method	URI	Description
POST	/v2/{tenant_id}/volumes	Creates a volume.
GET	/v2/{tenant_id}/volumes{?sort, limit, marker}	Lists summary information for all Block Storage volumes that the tenant can access.
GET	/v2/{tenant_id}/volumes/detail{?sort, limit, marker}	Lists all Block Storage volumes, with details, that the tenant can access.
GET	/v2/{tenant_id}/volumes/{volume_id}	Shows details for a volume.
PUT	/v2/{tenant_id}/volumes/{volume_id}	Updates a volume.
DELETE	/v2/{tenant_id}/volumes/{volume_id}	Deletes a volume.
POST	/v2/{tenant_id}/volumes/{volume_id}/metadata	Creates metadata for a volume.
GET	/v2/{tenant_id}/volumes/{volume_id}/metadata	Shows metadata for a volume.
PUT	/v2/{tenant_id}/volumes/{volume_id}/metadata	Updates metadata for a volume.

1.4.1. Create volume

Method	URI	Description
POST	/v2/{tenant_id}/volumes	Creates a volume.

To create a bootable volume, include the UUID of the image from which you want to create the volume in the `imageRef` attribute in the request body.

Preconditions

- You must have enough volume storage quota remaining to create a volume of size requested.

Asynchronous Postconditions

- With correct permissions, you can see the volume status as available through API calls.
- With correct access, you can see the created volume in the storage system that OpenStack Block Storage manages.

Troubleshooting

- If volume status remains `creating` or shows another error status, the request failed. Ensure you meet the preconditions then investigate the storage back end.
- Volume is not created in the storage system that OpenStack Block Storage manages.
- The storage node needs enough free storage space to match the size of the volume creation request.

Normal response codes: 202

1.4.1.1. Request

This table shows the URI parameters for the create volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.9. Create volume: JSON request

```
{
  "volume": {
    "size": 10,
    "availability_zone": null,
    "source_volid": null,
    "description": null,
    "multiattach ": false,
    "snapshot_id": null,
    "name": null,
    "imageRef": null,
    "volume_type": null,
    "metadata": {},
    "source_replica": null,
    "consistencygroup_id": null
  }
}
```

This table shows the body parameters for the create volume request:

Name	Type	Description
volume	Dict (Required)	A volume object.
size	Int (Required)	The size of the volume, in gibibytes (GiB).
availability_zone	String (Optional)	The availability zone.
source_volid	UUID (Optional)	The UUID of the source volume. The API creates a new volume with the same size as the source volume.
description	String (Optional)	The volume description.
multiattach	Boolean (Optional)	To enable this volume to attach to more than one server, set this value to <code>true</code> . Default is <code>false</code> .
snapshot_id	UUID (Optional)	To create a volume from an existing snapshot, specify the UUID of the volume snapshot. The volume is created in same availability zone and with same size as the snapshot.
name	String (Optional)	The volume name.
imageRef	UUID (Optional)	The UUID of the image from which you want to create the volume. Required to create a bootable volume.
volume_type	String (Optional)	The volume type. To create an environment with multiple-storage back ends, you must specify a volume type. Block Storage volume back ends are spawned as children to <code>cinder-volume</code> , and they are keyed from a unique queue. They are named <code>cinder-volume.HOST.BACKEND</code> . For example, <code>cinder-volume.ubuntu.lvmdriver</code> . When a volume is created, the scheduler chooses an appropriate back end to handle the request based on the volume type. Default is <code>None</code> . For information about how to use volume types to create multiple-storage back ends, see Configure multiple-storage back ends .
metadata	Dict (Optional)	One or more metadata key and value pairs that are associated with the volume.
source_replica	UUID (Optional)	The UUID of the primary volume to clone.
consistencygroup_id	UUID (Optional)	The UUID of the consistency group.
scheduler_hints	Dict (Optional)	The dictionary of data to send to the scheduler.

Example 1.10. Create volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume
  xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
  name="vol-001" description="Another volume."
```

```
size="2"/>
```

This operation does not accept a request body.

1.4.1.2. Response

Example 1.11. Create volume: JSON response

```
{
  "volume": {
    "status": "creating",
    "migration_status": null,
    "user_id": "0eea4eabcf184061a3b6db1e0daaf010",
    "attachments": [],
    "links": [
      {
        "href": "http://23.253.248.171:8776/v2/
bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/6edbc2f4-1507-44f8-ac0d-
eed1d2608d38",
        "rel": "self"
      },
      {
        "href": "http://23.253.248.171:8776/
bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/6edbc2f4-1507-44f8-ac0d-
eed1d2608d38",
        "rel": "bookmark"
      }
    ],
    "availability_zone": "nova",
    "bootable": "false",
    "encrypted": false,
    "created_at": "2015-11-29T03:01:44.000000",
    "description": null,
    "updated_at": null,
    "volume_type": "lvmdriver-1",
    "name": "test-volume-attachments",
    "replication_status": "disabled",
    "consistencygroup_id": null,
    "source_volid": null,
    "snapshot_id": null,
    "multiattach": false,
    "metadata": {},
    "id": "6edbc2f4-1507-44f8-ac0d-eed1d2608d38",
    "size": 2
  }
}
```

This table shows the body parameters for the create volume response:

Name	Type	Description
volume	Dict (Required)	A volume object.
status	String (Required)	The volume status.
migration_status	String (Required)	The volume migration status.
user_id	UUID	The UUID of the user.

Name	Type	Description
	(Required)	
attachments	List (Required)	Instance attachment information. If this volume is attached to a server instance, the attachments list includes the UUID of the attached server, an attachment UUID, the name of the attached host, if any, the volume UUID, the device, and the device UUID. Otherwise, this list is empty.
links	List (Required)	The volume links.
availability_zone	String (Required)	The availability zone.
bootable	Boolean (Required)	Enables or disables the bootable attribute. You can boot an instance from a bootable volume.
encrypted	Boolean (Required)	If true, this volume is encrypted.
created_at	DateTime (Required)	The date and time when the resource was created. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm For example, 2015-08-27T09:49:58-05:00. The ±hh:mm value, if included, is the time zone as an offset from UTC.
description	String (Required)	The volume description.
updated_at	DateTime (Required)	The date and time when the resource was updated. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm For example, 2015-08-27T09:49:58-05:00. The ±hh:mm value, if included, is the time zone as an offset from UTC. In the previous example, the offset value is -05:00. If the updated_at date and time stamp is not set, its value is null.
volume_type	String (Required)	The volume type. In an environment with multiple-storage back ends, the scheduler determines where to send the volume based on the volume type. For information about how to use volume types to create multiple-storage back ends, see Configure multiple-storage back ends .
name	String (Required)	The volume name.
replication_status	String (Required)	The volume replication status.
consistencygroup_id	UUID (Required)	The UUID of the consistency group.
source_volid	UUID (Required)	The UUID of the source volume.

Name	Type	Description
snapshot_id	UUID (Required)	The UUID of the source volume snapshot. The API creates a new volume snapshot with the same size as the source volume snapshot.
multiattach	Boolean (Required)	If true, this volume can attach to more than one instance.
metadata	Dict (Required)	One or more metadata key and value pairs that are associated with the volume.
id	UUID (Required)	The UUID of the volume.
size	Int (Required)	The size of the volume, in gibibytes (GiB).

Example 1.12. Create volume: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volume xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/volume/api/v1" status="creating"
  name="vol-001" availability_zone="nova" bootable="false"
  created_at="2014-02-21 20:18:33.122452"
  description="Another volume." volume_type="None"
  snapshot_id="None" source_volid="None"
  id="83960a54-8dad-4fd8-bc41-33c71e098e04" size="2">
  <attachments/>
  <metadata/>
</volume>
```

This operation does not return a response body.

1.4.2. List volumes

Method	URI	Description
GET	/v2/{tenant_id}/volumes{?sort, limit, marker}	Lists summary information for all Block Storage volumes that the tenant can access.

Normal response codes: 200

1.4.2.1. Request

This table shows the URI parameters for the list volumes request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list volumes request:

Name	Type	Description
sort	String (Optional)	Comma-separated list of sort keys and optional sort directions in the form of <key>[:<direction>]. A valid direction is asc (ascending) or desc (descending).
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

1.4.2.2. Response

Example 1.13. List volumes: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumes xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content">
  <volume name="vol-004" id="45baf976-c20a-4894-a7c3-c94b7376bf55">
    <attachments/>
    <metadata/>
  </volume>
  <volume name="vol-003" id="5aa119a8-d25b-45a7-8d1b-88e127885635">
    <attachments/>
    <metadata/>
  </volume>
</volumes>
```

Example 1.14. List volumes: JSON response

```
{
  "volumes": [
    {
```

```

        "id": "45baf976-c20a-4894-a7c3-c94b7376bf55",
        "links": [
            {
                "href": "http://localhost:8776/v2/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/45baf976-c20a-4894-a7c3-
c94b7376bf55",
                "rel": "self"
            },
            {
                "href": "http://localhost:8776/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/45baf976-c20a-4894-a7c3-
c94b7376bf55",
                "rel": "bookmark"
            }
        ],
        "name": "vol-004"
    },
    {
        "id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
        "links": [
            {
                "href": "http://localhost:8776/
v2/0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
                "rel": "self"
            },
            {
                "href": "http://localhost:8776/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
                "rel": "bookmark"
            }
        ],
        "name": "vol-003"
    }
]
}

```

This operation does not return a response body.

1.4.3. List volumes with details

Method	URI	Description
GET	/v2/{tenant_id}/volumes/detail{?sort,limit,marker}	Lists all Block Storage volumes, with details, that the tenant can access.

Normal response codes: 200

1.4.3.1. Request

This table shows the URI parameters for the list volumes with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list volumes with details request:

Name	Type	Description
sort	String (Optional)	Comma-separated list of sort keys and optional sort directions in the form of <key>[:<direction>]. A valid direction is asc (ascending) or desc (descending).
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

1.4.3.2. Response

Example 1.15. List volumes with details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumes
  xmlns:os-vol-image-meta="http://docs.openstack.org/openstack-block-storage/2.0/content/Volume_Image_Metadata.html"
  xmlns:os-vol-tenant-attr="http://docs.openstack.org/openstack-block-storage/2.0/content/Volume_Tenant_Attribute.html"
  xmlns:os-vol-host-attr="http://docs.openstack.org/openstack-block-storage/2.0/content/Volume_Host_Attribute.html"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content">
  <volume status="available" name="vol-004" availability_zone="nova"
    created_at="2013-02-25 06:36:28" description="Another volume."
    volume_type="None" source_vol_id="None" snapshot_id="None"
    id="45baf976-c20a-4894-a7c3-c94b7376bf55" size="1"
    os-vol-tenant-attr:tenant_id="0c2eba2c5af04d3f9e9d0d410b371fde"
    os-vol-host-attr:host="ip-10-168-107-25">
    <attachments/>
    <metadata>
      <meta key="contents">junk</meta>
    </metadata>
  </volume>
```

```

<volume status="available" name="vol-003" availability_zone="nova"
  created_at="2013-02-25 02:40:21"
  description="This is yet, another volume." volume_type="None"
  source_volid="None" snapshot_id="None"
  id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
  os-vol-tenant-attr:tenant_id="0c2eba2c5af04d3f9e9d0d410b371fde"
  os-vol-host-attr:host="ip-10-168-107-25">
  <attachments/>
  <metadata>
    <meta key="contents">not junk</meta>
  </metadata>
</volume>
</volumes>

```

Example 1.16. List volumes with details: JSON response

```

{
  "volumes": [
    {
      "migration_status": null,
      "attachments": [
        {
          "server_id": "f4fda93b-06e0-4743-8117-bc8bcecd651b",
          "attachment_id": "3b4db356-253d-4fab-bfa0-e3626c0b8405",
          "host_name": null,
          "volume_id": "6edbc2f4-1507-44f8-ac0d-eed1d2608d38",
          "device": "/dev/vdb",
          "id": "6edbc2f4-1507-44f8-ac0d-eed1d2608d38"
        }
      ],
      "links": [
        {
          "href": "http://23.253.248.171:8776/v2/
bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/6edbc2f4-1507-44f8-ac0d-
eed1d2608d38",
          "rel": "self"
        },
        {
          "href": "http://23.253.248.171:8776/
bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/6edbc2f4-1507-44f8-ac0d-
eed1d2608d38",
          "rel": "bookmark"
        }
      ],
      "availability_zone": "nova",
      "os-vol-host-attr:host": "difleming@lvmdriver-1#lvmdriver-1",
      "encrypted": false,
      "os-volume-replication:extended_status": null,
      "replication_status": "disabled",
      "snapshot_id": null,
      "id": "6edbc2f4-1507-44f8-ac0d-eed1d2608d38",
      "size": 2,
      "user_id": "32779452fcd34ae1a53a797ac8a1e064",
      "os-vol-tenant-attr:tenant_id":
"bab7d5c60cd041a0a36f7c4b6e1dd978",
      "os-vol-mig-status-attr:migstat": null,
      "metadata": {
        "readonly": false,
        "attached_mode": "rw"
      }
    }
  ],

```

```

        "status": "in-use",
        "description": null,
        "multiattach": true,
        "os-volume-replication:driver_data": null,
        "source_volid": null,
        "consistencygroup_id": null,
        "os-vol-mig-status-attr:name_id": null,
        "name": "test-volume-attachments",
        "bootable": "false",
        "created_at": "2015-11-29T03:01:44.000000",
        "volume_type": "lvmdriver-1"
    },
    {
        "migration_status": null,
        "attachments": [],
        "links": [
            {
                "href": "http://23.253.248.171:8776/
v2/bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/173f7b48-
c4c1-4e70-9acc-086b39073506",
                "rel": "self"
            },
            {
                "href": "http://23.253.248.171:8776/
bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/173f7b48-
c4c1-4e70-9acc-086b39073506",
                "rel": "bookmark"
            }
        ],
        "availability_zone": "nova",
        "os-vol-host-attr:host": "difleming@lvmdriver-1#lvmdriver-1",
        "encrypted": false,
        "os-volume-replication:extended_status": null,
        "replication_status": "disabled",
        "snapshot_id": null,
        "id": "173f7b48-c4c1-4e70-9acc-086b39073506",
        "size": 1,
        "user_id": "32779452fcd34ae1a53a797ac8a1e064",
        "os-vol-tenant-attr:tenant_id":
        "bab7d5c60cd041a0a36f7c4b6e1dd978",
        "os-vol-mig-status-attr:migstat": null,
        "metadata": {},
        "status": "available",
        "volume_image_metadata": {
            "kernel_id": "8a55f5f1-78f7-4477-8168-977d8519342c",
            "checksum": "eb9139e4942121f22bbc2afc0400b2a4",
            "min_ram": "0",
            "ramdisk_id": "5f6bdf8a-92db-4988-865b-60bdd808d9ef",
            "disk_format": "ami",
            "image_name": "cirros-0.3.4-x86_64-uec",
            "image_id": "b48c53e1-9a96-4a5a-a630-2e74ec54ddcc",
            "container_format": "ami",
            "min_disk": "0",
            "size": "25165824"
        },
        "description": "",
        "multiattach": false,
        "os-volume-replication:driver_data": null,
        "source_volid": null,
        "consistencygroup_id": null,

```

```
        "os-vol-mig-status-attr:name_id": null,  
        "name": "test-volume",  
        "bootable": "true",  
        "created_at": "2015-11-29T02:25:18.000000",  
        "volume_type": "lvmdriver-1"  
      }  
    ]  
  }
```

This operation does not return a response body.

1.4.4. Show volume details

Method	URI	Description
GET	/v2/{tenant_id}/volumes/{volume_id}	Shows details for a volume.

Preconditions

- The volume must exist.

Normal response codes: 200

1.4.4.1. Request

This table shows the URI parameters for the show volume details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

This operation does not accept a request body.

1.4.4.2. Response

Example 1.17. Show volume details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volume
  xmlns:os-vol-image-meta="http://docs.openstack.org/openstack-block-
storage/2.0/content/Volume_Image_Metadata.html"
  xmlns:os-vol-tenant-attr="http://docs.openstack.org/openstack-block-
storage/2.0/content/Volume_Tenant_Attribute.html"
  xmlns:os-vol-host-attr="http://docs.openstack.org/openstack-block-storage/
2.0/content/Volume_Host_Attribute.html"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content"
  status="available" name="vol-003" availability_zone="nova"
  bootable="false" created_at="2013-02-25 02:40:21"
  description="This is yet, another volume." volume_type="None"
  source_volid="None" snapshot_id="None"
  id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
  os-vol-tenant-attr:tenant_id="0c2eba2c5af04d3f9e9d0d410b371fde"
  os-vol-host-attr:host="ip-10-168-107-25">
  <attachments/>
  <metadata>
    <meta key="contents">not junk</meta>
  </metadata>
</volume>
```

Example 1.18. Show volume details: JSON response

```
{
  "volume": {
    "status": "available",
```

```
    "attachments": [],
    "links": [
      {
        "href": "http://localhost:8776/v2/0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-d25b-45a7-8d1b-88e127885635",
        "rel": "self"
      },
      {
        "href": "http://localhost:8776/v2/0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-d25b-45a7-8d1b-88e127885635",
        "rel": "bookmark"
      }
    ],
    "availability_zone": "nova",
    "bootable": "false",
    "os-vol-host-attr:host": "ip-10-168-107-25",
    "source_volid": null,
    "snapshot_id": null,
    "id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
    "description": "Super volume.",
    "name": "vol-002",
    "created_at": "2013-02-25T02:40:21.000000",
    "volume_type": "None",
    "os-vol-tenant-attr:tenant_id": "0c2eba2c5af04d3f9e9d0d410b371fde",
    "size": 1,
    "os-volume-replication:driver_data": null,
    "os-volume-replication:extended_status": null,
    "metadata": {
      "contents": "not junk"
    }
  }
}
```

This operation does not return a response body.

1.4.5. Update volume

Method	URI	Description
PUT	/v2/{tenant_id}/volumes/{volume_id}	Updates a volume.

Normal response codes: 200

1.4.5.1. Request

This table shows the URI parameters for the update volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

This table shows the body parameters for the update volume request:

Name	Type	Description
volume	Dict (Required)	A volume object.
name	String (Optional)	The volume name.
description	String (Optional)	The volume description.
metadata	Dict (Optional)	One or more metadata key and value pairs that are associated with the volume.

Example 1.19. Update volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot
  xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
  name="vol-003" description="This is yet, another volume."/>
```

Example 1.20. Update volume: JSON request

```
{
  "volume": {
    "name": "vol-003",
    "description": "This is yet, another volume."
  }
}
```

This operation does not accept a request body.

1.4.5.2. Response

Example 1.21. Update volume: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<volume xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content"
  status="available" name="vol-003" availability_zone="nova"
  created_at="2013-02-25 02:40:21"
  description="This is yet, another volume." volume_type="None"
  source_volid="None" snapshot_id="None"
  id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1">
  <attachments/>
  <metadata>
    <meta key="contents">not junk</meta>
  </metadata>
</volume>
```

Example 1.22. Update volume: JSON response

```
{
  "volume": {
    "status": "available",
    "migration_status": null,
    "user_id": "0eea4eabcf184061a3b6db1e0daaf010",
    "attachments": [],
    "links": [
      {
        "href": "http://localhost:8776/v2/0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-d25b-45a7-8d1b-88e127885635",
        "rel": "self"
      },
      {
        "href": "http://localhost:8776/v2/0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-d25b-45a7-8d1b-88e127885635",
        "rel": "bookmark"
      }
    ],
    "availability_zone": "nova",
    "bootable": "false",
    "encrypted": false,
    "created_at": "2015-11-29T03:01:44.000000",
    "description": "This is yet, another volume.",
    "updated_at": null,
    "volume_type": "lvmdriver-1",
    "name": "vol-003",
    "replication_status": "disabled",
    "consistencygroup_id": null,
    "source_volid": null,
    "snapshot_id": null,
    "multiattach": false,
    "metadata": {
      "contents": "not junk"
    },
    "id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
    "size": 1
  }
}
```

This operation does not return a response body.

1.4.6. Delete volume

Method	URI	Description
DELETE	/v2/{tenant_id}/volumes/{volume_id}	Deletes a volume.

Preconditions

- Volume status must be available, in-use, error, or error_restoring.
- You cannot already have a snapshot of the volume.
- You cannot delete a volume that is in a migration.

Asynchronous Postconditions

- The volume is deleted in volume index.
- The volume managed by OpenStack Block Storage is deleted in storage node.

Troubleshooting

- If volume status remains in deleting or becomes error_deleting the request failed. Ensure you meet the preconditions then investigate the storage back end.
- The volume managed by OpenStack Block Storage is not deleted from the storage system.

Normal response codes: 202

1.4.6.1. Request

This table shows the URI parameters for the delete volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

This operation does not accept a request body.

1.4.7. Create volume metadata

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/metadata	Creates metadata for a volume.

Normal response codes: 202

1.4.7.1. Request

This table shows the URI parameters for the create volume metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

Example 1.23. Create volume metadata: JSON request

```
{
  "metadata": {
    "name": "metadata0"
  }
}
```

1.4.7.2. Response

Example 1.24. Create volume metadata: JSON response

```
{
  "metadata": {
    "name": "metadata0"
  }
}
```

1.4.8. Show volume metadata

Method	URI	Description
GET	/v2/{tenant_id}/volumes/{volume_id}/metadata	Shows metadata for a volume.

Normal response codes: 200

1.4.8.1. Request

This table shows the URI parameters for the show volume metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

This operation does not accept a request body.

1.4.8.2. Response

Example 1.25. Show volume metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1"/>
```

Example 1.26. Show volume metadata: JSON response

```
{
  "metadata": {}
}
```

This operation does not return a response body.

1.4.9. Update volume metadata

Method	URI	Description
PUT	/v2/{tenant_id}/volumes/{volume_id}/metadata	Updates metadata for a volume.

Replaces metadata items that match keys. Does not modify items that are not in the request.

Normal response codes: 200

1.4.9.1. Request

This table shows the URI parameters for the update volume metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

Example 1.27. Update volume metadata: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata>
  <meta key="key">v2</meta>
</metadata>
```

Example 1.28. Update volume metadata: JSON request

```
{
  "metadata": {
    "name": "metadata1"
  }
}
```

This operation does not accept a request body.

1.4.9.2. Response

Example 1.29. Update volume metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
  <meta key="key">v2</meta>
</metadata>
```

Example 1.30. Update volume metadata: JSON response

```
{
  "metadata": {
    "name": "metadata1"
  }
}
```

This operation does not return a response body.

1.5. Volume type access (volumes)

Private volume type access to project.

By default, volumes types are public. To create a private volume type, set the `is_public` boolean field to `false` at volume type creation time. To control access to a private volume type, user needs to add a project to or remove a project from the volume type. Private volume types without projects are only accessible by users with the administrative role and context.

Method	URI	Description
POST	/v2/{tenant_id}/types/{volume_type}/action	Adds private volume type access to a project.
POST	/v2/{tenant_id}/types/{volume_type}/action	Removes private volume type access from a project.
GET	/v2/{tenant_id}/types/{volume_type}/os-volume-type-access	Lists project IDs that have access to private volume type.

1.5.1. Add private volume type access

Method	URI	Description
POST	/v2/{tenant_id}/types/{volume_type}/action	Adds private volume type access to a project.

Normal response codes: 202

1.5.1.1. Request

This table shows the URI parameters for the add private volume type access request:

Name	Type	Description
{tenant_id}	UUID	The ID of the project. Volume Type access to be added to this project ID.
{volume_type}	UUID	The ID of Volume Type to be accessed by project.

Example 1.31. Add private volume type access: JSON request

```
{
  "addProjectAccess": {
    "project": "f270b245cb11498ca4031deb7e141cfa"
  }
}
```

Example 1.32. Add private volume type access: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<addProjectAccess id="bb4f8f7f-fc38-4807-bd78-5710792205e1">
  <project>"f270b245cb11498ca4031deb7e141cfa"</project>
</addProjectAccess>
```

This operation does not accept a request body.

1.5.2. Remove private volume type access

Method	URI	Description
POST	/v2/{tenant_id}/types/{volume_type}/action	Removes private volume type access from a project.

Normal response codes: 202

1.5.2.1. Request

This table shows the URI parameters for the remove private volume type access request:

Name	Type	Description
{tenant_id}	UUID	The ID of the project. Volume Type access to be added to this project ID.
{volume_type}	UUID	The ID of Volume Type to be accessed by project.

Example 1.33. Remove private volume type access: JSON request

```
{
  "removeProjectAccess": {
    "project": "f270b245cb11498ca4031deb7e141cfa"
  }
}
```

Example 1.34. Remove private volume type access: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<removeProjectAccess id="bb4f8f7f-fc38-4807-bd78-5710792205e1">
  <project>"f270b245cb11498ca4031deb7e141cfa"</project>
</removeProjectAccess>
```

This operation does not accept a request body.

1.5.3. List private volume type access details

Method	URI	Description
GET	/v2/{tenant_id}/types/{volume_type}/os-volume-type-access	Lists project IDs that have access to private volume type.

Normal response codes: 200

1.5.3.1. Request

This table shows the URI parameters for the list private volume type access details request:

Name	Type	Description
{tenant_id}	UUID	The ID of the project. Volume Type access to be added to this project ID.
{volume_type}	UUID	The ID of Volume Type to be accessed by project.

1.5.3.2. Response

Example 1.35. List private volume type access details: JSON response

```
{
  "volume_type_access": {
    "volume_type_id": "3c67e124-39ad-4ace-a507-8bb7bf510c26",
    "project_id": "f270b245cb11498ca4031deb7e141cfa"
  }
}
```

1.6. Volume actions (volumes, action)

Extends the size of, resets statuses for, sets image metadata for, and removes image metadata from a volume. Attaches a volume to a server, detaches a volume from a server, and removes a volume from Block Storage management without actually removing the back-end storage object associated with it.

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Extends the size of a volume to a requested size, in gibibytes (GiB). Specify the os-extend action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Resets the status, attach status, and migration status for a volume. Specify the os-reset_status action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Sets the image metadata for a volume. Specify the os-set_image_metadata action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Removes image metadata, by key, from a volume. Specify the os-unset_image_metadata action in the request body and the key for the metadata key and value pair that you want to remove.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Attaches a volume to a server. Specify the os-attach action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Removes a volume from Block Storage management without removing the back-end storage object that is associat-

Method	URI	Description
		ed with it. Specify the os-unmanage action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Forces a volume to detach. Specify the os-force_detach action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Promotes a replicated volume. Specify the os-promote-replica action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Re-enables replication of a volume. Specify the volume-replica-reenable action in the request body.

1.6.1. Extend volume size

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Extends the size of a volume to a requested size, in gibibytes (GiB). Specify the os-extend action in the request body.

Preconditions

- Volume status must be available.
- Sufficient amount of storage must exist to extend the volume.
- The user quota must have sufficient volume storage.

Troubleshooting

- An error_extending volume status indicates that the request failed. Ensure that you meet the preconditions and retry the request. If the request fails again, investigate the storage back end.

Normal response codes: 202

1.6.1.1. Request

This table shows the URI parameters for the extend volume size request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

Example 1.36. Extend volume size: JSON request

```
{
  "os-extend": {
    "new_size": 3
  }
}
```

1.6.2. Reset volume statuses

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Resets the status, attach status, and migration status for a volume. Specify the os-reset_status action in the request body.

Normal response codes: 202

1.6.2.1. Request

This table shows the URI parameters for the reset volume statuses request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

Example 1.37. Reset volume statuses: JSON request

```
{
  "os-reset_status": {
    "status": "available",
    "attach_status": "detached",
    "migration_status": "migrating"
  }
}
```

1.6.3. Set image metadata for volume

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Sets the image metadata for a volume. Specify the os-set_image_metadata action in the request body.

Normal response codes: 202

1.6.3.1. Request

This table shows the URI parameters for the set image metadata for volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

Example 1.38. Set image metadata for volume: JSON request

```
{
  "os-set_image_metadata": {
    "metadata": {
      "image_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
      "image_name": "image",
      "kernel_id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
      "ramdisk_id": "somedisk"
    }
  }
}
```

1.6.4. Remove image metadata from volume

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Removes image metadata, by key, from a volume. Specify the os-unset_image_metadata action in the request body and the key for the metadata key and value pair that you want to remove.

Normal response codes: 202

1.6.4.1. Request

This table shows the URI parameters for the remove image metadata from volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

Example 1.39. Remove image metadata from volume: JSON request

```
{
  "os-unset_image_metadata": {
    "key": "ramdisk_id"
  }
}
```

1.6.5. Attach volume to server

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Attaches a volume to a server. Specify the os-attach action in the request body.

Preconditions

- Volume status must be available.
- You should set instance_uuid or host_name.

Normal response codes: 202

1.6.5.1. Request

This table shows the URI parameters for the attach volume to server request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

Example 1.40. Attach volume to server: JSON request

```
{
  "os-attach": {
    "instance_uuid": "95D9EF50-507D-11E5-B970-0800200C9A66",
    "mountpoint": "/dev/vdc"
  }
}
```

1.6.6. Unmanage volume

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Removes a volume from Block Storage management without removing the back-end storage object that is associated with it. Specify the os-unmanage action in the request body.

Preconditions

- Volume status must be available.

Normal response codes: 202

1.6.6.1. Request

This table shows the URI parameters for the unmanage volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

Example 1.41. Unmanage volume: JSON request

```
{
  "os-unmanage": {}
}
```

1.6.7. Force detach volume

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Forces a volume to detach. Specify the os-force_detach action in the request body.

Rolls back an unsuccessful detach operation after you disconnect the volume.

Normal response codes: 202

1.6.7.1. Request

This table shows the URI parameters for the force detach volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

Example 1.42. Force detach volume: JSON request

```
{
  "os-force_detach": {
    "attachment_id": "d8777f54-84cf-4809-a679-468ffed56cf1",
    "connector": {
      "initiator": "iqn.2012-07.org.fake:01"
    }
  }
}
```


1.6.8. Promote replicated volume

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Promotes a replicated volume. Specify the os-promote-replica action in the request body.

Normal response codes: 202

1.6.8.1. Request

This table shows the URI parameters for the promote replicated volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

Example 1.43. Promote replicated volume: JSON request

```
{
  "os-promote-replica": {}
}
```

1.6.9. Reenable volume replication

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Re-enables replication of a volume. Specify the volume-replica-reenable action in the request body.

Normal response codes: 202

1.6.9.1. Request

This table shows the URI parameters for the reenable volume replication request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

Example 1.44. Reenable volume replication: JSON request

```
{
  "os-reenable-replica": {}
}
```

1.7. Backups (backups)

A backup is a full copy of a volume stored in an external service. The service can be configured. The only supported service is Object Storage. A backup can subsequently be restored from the external service to either the same volume that the backup was originally taken from or to a new volume. Backup and restore operations can only be carried out on volumes that are in an unattached and available state.

When you create, list, or delete backups, these status values are possible:

Table 1.2. Backup statuses

Status	Description
creating	The backup is being created.
available	The backup is ready to restore to a volume.
deleting	The backup is being deleted.
error	A backup error occurred.
restoring	The backup is being restored to a volume.
error_restoring	A backup restoration error occurred.

If an error occurs, you can find more information about the error in the `fail_reason` field for the backup.

Method	URI	Description
POST	/v2/{tenant_id}/backups	Creates a Block Storage backup from a volume.
GET	/v2/{tenant_id}/backups{?sort_key, sort_dir, limit, marker}	Lists Block Storage backups to which the tenant has access.

Method	URI	Description
GET	/v2/{tenant_id}/backups/detail{?sort_key,sort_dir,limit,marker}	Lists Block Storage backups, with details, to which the tenant has access.
GET	/v2/{tenant_id}/backups/{backup_id}	Shows details for a backup.
DELETE	/v2/{tenant_id}/backups/{backup_id}	Deletes a backup.
POST	/v2/{tenant_id}/backups/{backup_id}/restore	Restores a Block Storage backup to an existing or new Block Storage volume.

1.7.1. Create backup

Method	URI	Description
POST	/v2/{tenant_id}/backups	Creates a Block Storage backup from a volume.

Normal response codes: 202

1.7.1.1. Request

This table shows the URI parameters for the create backup request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.45. Create backup: JSON request

```
{
  "backup": {
    "container": null,
    "description": null,
    "name": "backup001",
    "volume_id": "64f5d2fb-d836-4063-b7e2-544d5c1ff607",
    "incremental": true
  }
}
```

1.7.1.2. Response

Example 1.46. Create backup: JSON response

```
{
  "backup": {
    "id": "deac8b8c-35c9-4c71-acaa-889c2d5d5c8e",
    "links": [
      {
        "href": "http://localhost:8776/v2/c95fc3e4afe248a49a28828f286a7b38/backups/deac8b8c-35c9-4c71-acaa-889c2d5d5c8e",
        "rel": "self"
      },
      {
        "href": "http://localhost:8776/v2/c95fc3e4afe248a49a28828f286a7b38/backups/deac8b8c-35c9-4c71-acaa-889c2d5d5c8e",
        "rel": "bookmark"
      }
    ],
    "name": "backup001"
  }
}
```

1.7.2. List backups

Method	URI	Description
GET	/v2/{tenant_id}/backups{?sort_key,sort_dir,limit,marker}	Lists Block Storage backups to which the tenant has access.

Normal response codes: 200

1.7.2.1. Request

This table shows the URI parameters for the list backups request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list backups request:

Name	Type	Description
sort_key	String (Optional)	Sorts by an attribute. A valid value is name, status, container_format, disk_format, size, id, created_at, or updated_at. Default is created_at. The API uses the natural sorting direction of the sort_key attribute value.
sort_dir	String (Optional)	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is desc.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

1.7.2.2. Response

Example 1.47. List backups: JSON response

```
{
  "backups": [
    {
      "id": "2ef47aee-8844-490c-804d-2a8efe561c65",
      "links": [
        {
          "href": "http://localhost:8776/v1/c95fc3e4afe248a49a28828f286a7b38/backups/2ef47aee-8844-490c-804d-2a8efe561c65",
          "rel": "self"
        },
        {
          "href": "http://localhost:8776/c95fc3e4afe248a49a28828f286a7b38/backups/2ef47aee-8844-490c-804d-2a8efe561c65",

```

```
        "rel": "bookmark"
      }
    ],
    "name": "backup001"
  },
  {
    "id": "4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
    "links": [
      {
        "href": "http://localhost:8776/
v1/c95fc3e4afe248a49a28828f286a7b38/backups/
4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
        "rel": "self"
      },
      {
        "href": "http://localhost:8776/
c95fc3e4afe248a49a28828f286a7b38/backups/
4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
        "rel": "bookmark"
      }
    ],
    "name": "backup002"
  }
]
```

1.7.3. List backups with details

Method	URI	Description
GET	/v2/{tenant_id}/backups/detail{?sort_key,sort_dir,limit,marker}	Lists Block Storage backups, with details, to which the tenant has access.

Normal response codes: 200

1.7.3.1. Request

This table shows the URI parameters for the list backups with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list backups with details request:

Name	Type	Description
sort_key	String (Optional)	Sorts by an attribute. A valid value is name, status, container_format, disk_format, size, id, created_at, or updated_at. Default is created_at. The API uses the natural sorting direction of the sort_key attribute value.
sort_dir	String (Optional)	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is desc.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

1.7.3.2. Response

Example 1.48. List backups with details: JSON response

```
{
  "backups": [
    {
      "availability_zone": "az1",
      "container": "volumebackups",
      "created_at": "2013-04-02T10:35:27.000000",
      "description": null,
      "fail_reason": null,
      "id": "2ef47aee-8844-490c-804d-2a8efe561c65",
      "links": [
        {
          "href": "http://localhost:8776/v1/c95fc3e4afe248a49a28828f286a7b38/backups/2ef47aee-8844-490c-804d-2a8efe561c65",
          "rel": "self"
        }
      ]
    }
  ]
}
```

```

        },
        {
            "href": "http://localhost:8776/
c95fc3e4afe248a49a28828f286a7b38/backups/
2ef47aee-8844-490c-804d-2a8efe561c65",
            "rel": "bookmark"
        }
    ],
    "name": "backup001",
    "object_count": 22,
    "size": 1,
    "status": "available",
    "volume_id": "e5185058-943a-4cb4-96d9-72c184c337d6",
    "is_incremental": true,
    "has_dependent_backups": false
},
{
    "availability_zone": "az1",
    "container": "volumebackups",
    "created_at": "2013-04-02T10:21:48.000000",
    "description": null,
    "fail_reason": null,
    "id": "4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
    "links": [
        {
            "href": "http://localhost:8776/
v1/c95fc3e4afe248a49a28828f286a7b38/backups/
4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
            "rel": "self"
        },
        {
            "href": "http://localhost:8776/
c95fc3e4afe248a49a28828f286a7b38/backups/
4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
            "rel": "bookmark"
        }
    ],
    "name": "backup002",
    "object_count": 22,
    "size": 1,
    "status": "available",
    "volume_id": "e5185058-943a-4cb4-96d9-72c184c337d6",
    "is_incremental": true,
    "has_dependent_backups": false
}
]
}

```


1.7.4. Show backup details

Method	URI	Description
GET	/v2/{tenant_id}/backups/{backup_id}	Shows details for a backup.

Normal response codes: 200

1.7.4.1. Request

This table shows the URI parameters for the show backup details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{backup_id}	UUID	The UUID for a backup.

This operation does not accept a request body.

1.7.4.2. Response

Example 1.49. Show backup details: JSON response

```
{
  "backup": {
    "availability_zone": "az1",
    "container": "volumebackups",
    "created_at": "2013-04-02T10:35:27.000000",
    "description": null,
    "fail_reason": null,
    "id": "2ef47aee-8844-490c-804d-2a8efe561c65",
    "links": [
      {
        "href": "http://localhost:8776/v1/c95fc3e4afe248a49a28828f286a7b38/backups/2ef47aee-8844-490c-804d-2a8efe561c65",
        "rel": "self"
      },
      {
        "href": "http://localhost:8776/v1/c95fc3e4afe248a49a28828f286a7b38/backups/2ef47aee-8844-490c-804d-2a8efe561c65",
        "rel": "bookmark"
      }
    ],
    "name": "backup001",
    "object_count": 22,
    "size": 1,
    "status": "available",
    "volume_id": "e5185058-943a-4cb4-96d9-72c184c337d6",
    "is_incremental": true,
    "has_dependent_backups": false
  }
}
```

1.7.5. Delete backup

Method	URI	Description
DELETE	/v2/{tenant_id}/backups/{backup_id}	Deletes a backup.

Normal response codes: 204

1.7.5.1. Request

This table shows the URI parameters for the delete backup request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{backup_id}	UUID	The UUID for a backup.

This operation does not accept a request body.

1.7.6. Restore backup

Method	URI	Description
POST	/v2/{tenant_id}/backups/{backup_id}/restore	Restores a Block Storage backup to an existing or new Block Storage volume.

You must specify either the UUID or name of the volume. If you specify both the UUID and name, the UUID takes priority.

Normal response codes: 202

1.7.6.1. Request

This table shows the URI parameters for the restore backup request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{backup_id}	UUID	The UUID for a backup.

Example 1.50. Restore backup: JSON request

```
{
  "restore": {
    "name": "vol-01",
    "volume_id": "64f5d2fb-d836-4063-b7e2-544d5c1ff607"
  }
}
```

1.7.6.2. Response

Example 1.51. Restore backup: JSON response

```
{
  "restore": {
    "backup_id": "2ef47aee-8844-490c-804d-2a8efe561c65",
    "volume_id": "795114e8-7489-40be-a978-83797f2c1dd3"
  }
}
```

1.8. Backup actions (backups, action)

Force-deletes a backup.

Method	URI	Description
POST	/v2/{tenant_id}/backups/{backup_id}/action	Force-deletes a backup. Specify the os-force_delete action in the request body.

1.8.1. Force-delete backup

Method	URI	Description
POST	/v2/{tenant_id}/backups/{backup_id}/action	Force-deletes a backup. Specify the os-force_delete action in the request body.

This operations deletes the backup and any backup data.

The backup driver returns the 405 status code if it does not support this operation.

Normal response codes: 202

Error response codes: 404, 405

1.8.1.1. Request

This table shows the URI parameters for the force-delete backup request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{backup_id}	UUID	The UUID for a backup.

Example 1.52. Force-delete backup: JSON request

```
{
  "os-force_delete": {}
}
```

1.9. Capabilities for storage back ends (capabilities)

Shows capabilities for a storage back end.

Method	URI	Description
GET	/v2/{tenant_id}/capabilities/{hostname}	Shows capabilities for a storage back end.

1.9.1. Show back-end capabilities

Method	URI	Description
GET	/v2/{tenant_id}/capabilities/{hostname}	Shows capabilities for a storage back end.

Normal response codes: 200

1.9.1.1. Request

This table shows the URI parameters for the show back-end capabilities request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{hostname}	String	The name of the host that hosts the storage back end.

1.9.1.2. Response

Example 1.53. Show back-end capabilities: JSON response

```
{
  "namespace": "OS::Storage::Capabilities::fake",
  "vendor_name": "OpenStack",
  "volume_backend_name": "lvm",
  "pool_name": "pool",
  "driver_version": "2.0.0",
  "storage_protocol": "iSCSI",
  "display_name": "Capabilities of Cinder LVM driver",
  "description": "These are volume type options provided by Cinder LVM
driver, blah, blah.",
  "visibility": "public",
  "properties": {
    "compression": {
      "title": "Compression",
      "description": "Enables compression.",
      "type": "boolean"
    },
    "qos": {
      "title": "QoS",
      "description": "Enables QoS.",
      "type": "boolean"
    },
    "replication": {
      "title": "Replication",
      "description": "Enables replication.",
      "type": "boolean"
    },
    "thin_provisioning": {
      "title": "Thin Provisioning",
      "description": "Sets thin provisioning.",
      "type": "boolean"
    }
  }
}
```

This operation does not return a response body.

1.10. Quota sets extension (os-quota-sets)

Administrators only, depending on policy settings.

Shows, updates, and deletes quotas for a tenant.

Method	URI	Description
GET	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}{?usage}	Shows quotas for a tenant.
PUT	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.
DELETE	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}	Deletes quotas for a tenant so the quotas revert to default values.
GET	/v2/{tenant_id}/os-quota-sets/default	Gets default quotas for a tenant.
GET	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a tenant and user.
PUT	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a tenant and user.
DELETE	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Deletes quotas for a user so that the quotas revert to default values.
GET	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}	Shows details for quotas for a tenant and user.

1.10.1. Show quotas

Method	URI	Description
GET	/v2/{admin_tenant_id}/os-quotas-sets/{tenant_id}{?usage}	Shows quotas for a tenant.

Normal response codes: 200

1.10.1.1. Request

This table shows the URI parameters for the show quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.

This table shows the query parameters for the show quotas request:

Name	Type	Description
usage	Boolean (Optional)	Set to usage=true to show quota usage. Default is false.

This operation does not accept a request body.

1.10.1.2. Response

Example 1.54. Show quotas response: JSON

```
{
  "quota_set": {
    "gigabytes": 5,
    "snapshots": 10,
    "volumes": 20
  }
}
```

This table shows the body parameters for the show quotas response:

Name	Type	Description
quota_set	Dict (Required)	A quota_set object.
cores	Int (Required)	The number of instance cores that are allowed for each tenant.
fixed_ips	Int (Required)	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int (Required)	The number of floating IP addresses that are allowed for each tenant.
id	Int (Required)	The ID for the quota set.

Name	Type	Description
injected_file_content_bytes	Int (Required)	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes that are allowed for each injected file path.
injected_files	Int (Required)	The number of injected files that are allowed for each tenant.
instances	Int (Required)	The number of instances that are allowed for each tenant.
key_pairs	Int (Required)	The number of key pairs that are allowed for each user.
metadata_items	Int (Required)	The number of metadata items that are allowed for each instance.
ram	Int (Required)	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int (Optional)	The number of rules that are allowed for each security group.
security_groups	Int (Required)	The number of security groups that are allowed for each tenant.
in_use	String (Optional)	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int (Optional)	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 1.55. Show quotas response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <gigabytes>5</gigabytes>
  <snapshots>10</snapshots>
  <volumes>20</volumes>
</quota_set>
```

This operation does not return a response body.

1.10.2. Update quotas

Method	URI	Description
PUT	/v2/{admin_tenant_id}/os-quotas-sets/{tenant_id}	Updates quotas for a tenant.

Normal response codes: 200

1.10.2.1. Request

This table shows the URI parameters for the update quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.

Example 1.56. Update quotas response: JSON

```
{
  "quota_set": {
    "snapshots": 45
  }
}
```

This table shows the body parameters for the update quotas request:

Name	Type	Description
quota_set	Dict (Required)	A quota_set object.
cores	Int (Required)	The number of instance cores that are allowed for each tenant.
fixed_ips	Int (Required)	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int (Required)	The number of floating IP addresses that are allowed for each tenant.
id	Int (Required)	The ID for the quota set.
injected_file_content_bytes	Int (Required)	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes that are allowed for each injected file path.
injected_files	Int (Required)	The number of injected files that are allowed for each tenant.
instances	Int (Required)	The number of instances that are allowed for each tenant.
key_pairs	Int	The number of key pairs that are allowed for each user.

Name	Type	Description
	<i>(Required)</i>	
metadata_items	Int <i>(Required)</i>	The number of metadata items that are allowed for each instance.
ram	Int <i>(Required)</i>	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int <i>(Optional)</i>	The number of rules that are allowed for each security group.
security_groups	Int <i>(Required)</i>	The number of security groups that are allowed for each tenant.
in_use	String <i>(Optional)</i>	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int <i>(Optional)</i>	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 1.57. Show quotas response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <snapshots>45</snapshots>
</quota_set>
```

This operation does not accept a request body.

1.10.2.2. Response

Example 1.58. Update quota response: JSON

```
{
  "quota_set": {
    "snapshots": 45
  }
}
```

This table shows the body parameters for the update quotas response:

Name	Type	Description
quota_set	Dict <i>(Required)</i>	A quota_set object.
cores	Int <i>(Required)</i>	The number of instance cores that are allowed for each tenant.
fixed_ips	Int <i>(Required)</i>	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int <i>(Required)</i>	The number of floating IP addresses that are allowed for each tenant.
id	Int <i>(Required)</i>	The ID for the quota set.
injected_file_content_bytes	Int	The number of bytes of content that are allowed for each injected file.

Name	Type	Description
	<i>(Required)</i>	
injected_file_path_bytes	Int <i>(Required)</i>	The number of bytes that are allowed for each injected file path.
injected_files	Int <i>(Required)</i>	The number of injected files that are allowed for each tenant.
instances	Int <i>(Required)</i>	The number of instances that are allowed for each tenant.
key_pairs	Int <i>(Required)</i>	The number of key pairs that are allowed for each user.
metadata_items	Int <i>(Required)</i>	The number of metadata items that are allowed for each instance.
ram	Int <i>(Required)</i>	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int <i>(Optional)</i>	The number of rules that are allowed for each security group.
security_groups	Int <i>(Required)</i>	The number of security groups that are allowed for each tenant.
in_use	String <i>(Optional)</i>	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int <i>(Optional)</i>	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 1.59. Update quota response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <gigabytes>5</gigabytes>
  <snapshots>10</snapshots>
  <volumes>20</volumes>
</quota_set>
```

This operation does not return a response body.

1.10.3. Delete quotas

Method	URI	Description
DELETE	/v2/{admin_tenant_id}/os-quotas/{tenant_id}	Deletes quotas for a tenant so the quotas revert to default values.

Normal response codes: 200

1.10.3.1. Request

This table shows the URI parameters for the delete quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.

This operation does not accept a request body.

1.10.4. Get default quotas

Method	URI	Description
GET	/v2/{tenant_id}/os-quota-sets/default	Gets default quotas for a tenant.

Normal response codes: 200

1.10.4.1. Request

This table shows the URI parameters for the get default quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.10.4.2. Response

Example 1.60. Get default quotas response: JSON

```
{
  "quota_set": {
    "gigabytes": 5,
    "snapshots": 10,
    "volumes": 20
  }
}
```

This table shows the body parameters for the get default quotas response:

Name	Type	Description
quota_set	Dict (Required)	A quota_set object.
cores	Int (Required)	The number of instance cores that are allowed for each tenant.
fixed_ips	Int (Required)	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int (Required)	The number of floating IP addresses that are allowed for each tenant.
id	Int (Required)	The ID for the quota set.
injected_file_content_bytes	Int (Required)	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes that are allowed for each injected file path.
injected_files	Int	The number of injected files that are allowed for each tenant.

Name	Type	Description
	<i>(Required)</i>	
instances	Int <i>(Required)</i>	The number of instances that are allowed for each tenant.
key_pairs	Int <i>(Required)</i>	The number of key pairs that are allowed for each user.
metadata_items	Int <i>(Required)</i>	The number of metadata items that are allowed for each instance.
ram	Int <i>(Required)</i>	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int <i>(Optional)</i>	The number of rules that are allowed for each security group.
security_groups	Int <i>(Required)</i>	The number of security groups that are allowed for each tenant.
in_use	String <i>(Optional)</i>	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int <i>(Optional)</i>	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 1.61. Get default quotas response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <gigabytes>5</gigabytes>
  <snapshots>10</snapshots>
  <volumes>20</volumes>
</quota_set>
```

This operation does not return a response body.

1.10.5. Show quotas for user

Method	URI	Description
GET	/v2/{admin_tenant_id}/os-quotas-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a tenant and user.

Normal response codes: 200

1.10.5.1. Request

This table shows the URI parameters for the show quotas for user request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{user_id}	String	The user ID. Specify in the URI as user_id={user_id}.

This operation does not accept a request body.

1.10.5.2. Response

Example 1.62. Show quotas for user response: JSON

```
{
  "quota_set": {
    "snapshots": 45
  }
}
```

This table shows the body parameters for the show quotas for user response:

Name	Type	Description
quota_set	Dict (Required)	A quota_set object.
cores	Int (Required)	The number of instance cores that are allowed for each tenant.
fixed_ips	Int (Required)	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int (Required)	The number of floating IP addresses that are allowed for each tenant.
id	Int (Required)	The ID for the quota set.
injected_file_content_bytes	Int (Required)	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes that are allowed for each injected file path.
injected_files	Int	The number of injected files that are allowed for each tenant.

Name	Type	Description
	<i>(Required)</i>	
instances	Int <i>(Required)</i>	The number of instances that are allowed for each tenant.
key_pairs	Int <i>(Required)</i>	The number of key pairs that are allowed for each user.
metadata_items	Int <i>(Required)</i>	The number of metadata items that are allowed for each instance.
ram	Int <i>(Required)</i>	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int <i>(Optional)</i>	The number of rules that are allowed for each security group.
security_groups	Int <i>(Required)</i>	The number of security groups that are allowed for each tenant.
in_use	String <i>(Optional)</i>	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int <i>(Optional)</i>	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 1.63. Show quotas for user response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <gigabytes>5</gigabytes>
  <snapshots>10</snapshots>
  <volumes>20</volumes>
</quota_set>
```

This operation does not return a response body.

1.10.6. Update quotas for user

Method	URI	Description
PUT	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a tenant and user.

Normal response codes: 200

1.10.6.1. Request

This table shows the URI parameters for the update quotas for user request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{user_id}	String	The user ID. Specify in the URI as user_id={user_id}.

Example 1.64. Update quotas for user request: JSON

```
{
  "quota_set": {
    "snapshots": 45
  }
}
```

This table shows the body parameters for the update quotas for user request:

Name	Type	Description
quota_set	Dict (Required)	A quota_set object.
cores	Int (Required)	The number of instance cores that are allowed for each tenant.
fixed_ips	Int (Required)	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int (Required)	The number of floating IP addresses that are allowed for each tenant.
id	Int (Required)	The ID for the quota set.
injected_file_content_bytes	Int (Required)	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes that are allowed for each injected file path.
injected_files	Int (Required)	The number of injected files that are allowed for each tenant.
instances	Int (Required)	The number of instances that are allowed for each tenant.
key_pairs	Int	The number of key pairs that are allowed for each user.

Name	Type	Description
	<i>(Required)</i>	
metadata_items	Int <i>(Required)</i>	The number of metadata items that are allowed for each instance.
ram	Int <i>(Required)</i>	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int <i>(Optional)</i>	The number of rules that are allowed for each security group.
security_groups	Int <i>(Required)</i>	The number of security groups that are allowed for each tenant.
in_use	String <i>(Optional)</i>	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int <i>(Optional)</i>	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 1.65. Update quotas for user request: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <snapshots>45</snapshots>
</quota_set>
```

This operation does not accept a request body.

1.10.6.2. Response

Example 1.66. Update quotas for user response: JSON

```
{
  "quota_set": {
    "snapshots": 45
  }
}
```

This table shows the body parameters for the update quotas for user response:

Name	Type	Description
quota_set	Dict <i>(Required)</i>	A quota_set object.
cores	Int <i>(Required)</i>	The number of instance cores that are allowed for each tenant.
fixed_ips	Int <i>(Required)</i>	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int <i>(Required)</i>	The number of floating IP addresses that are allowed for each tenant.
id	Int <i>(Required)</i>	The ID for the quota set.
injected_file_content_bytes	Int	The number of bytes of content that are allowed for each injected file.

Name	Type	Description
	<i>(Required)</i>	
injected_file_path_bytes	Int <i>(Required)</i>	The number of bytes that are allowed for each injected file path.
injected_files	Int <i>(Required)</i>	The number of injected files that are allowed for each tenant.
instances	Int <i>(Required)</i>	The number of instances that are allowed for each tenant.
key_pairs	Int <i>(Required)</i>	The number of key pairs that are allowed for each user.
metadata_items	Int <i>(Required)</i>	The number of metadata items that are allowed for each instance.
ram	Int <i>(Required)</i>	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int <i>(Optional)</i>	The number of rules that are allowed for each security group.
security_groups	Int <i>(Required)</i>	The number of security groups that are allowed for each tenant.
in_use	String <i>(Optional)</i>	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int <i>(Optional)</i>	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 1.67. Show quotas for user response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <gigabytes>5</gigabytes>
  <snapshots>10</snapshots>
  <volumes>20</volumes>
</quota_set>
```

This operation does not return a response body.

1.10.7. Delete quotas for user

Method	URI	Description
DELETE	/v2/{admin_tenant_id}/os-quotas/{tenant_id}/{user_id}	Deletes quotas for a user so that the quotas revert to default values.

Normal response codes: 200

1.10.7.1. Request

This table shows the URI parameters for the delete quotas for user request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{user_id}	String	The user ID. Specify in the URI as user_id={user_id}.

This operation does not accept a request body.

1.10.8. Show quota details for user

Method	URI	Description
GET	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}	Shows details for quotas for a tenant and user.

Normal response codes: 200

1.10.8.1. Request

This table shows the URI parameters for the show quota details for user request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{user_id}	String	The user ID. Specify in the URI as user_id={user_id}.

This operation does not accept a request body.

1.10.8.2. Response

Example 1.68. Show quota details for user response: JSON

```
{
  "quota_set": {
    "snapshots": 45
  }
}
```

1.11. Quality of service (QoS) specifications (qos-specs)

Administrators only, depending on policy settings.

Creates, lists, shows details for, associates, disassociates, sets keys, unsets keys, and deletes quality of service (QoS) specifications.

Method	URI	Description
POST	/v2/{tenant_id}/qos-specs	Creates a QoS specification.
GET	/v2/{tenant_id}/qos-specs{?sort_key,sort_dir,limit,marker}	Lists quality of service (QoS) specifications.
GET	/v2/{tenant_id}/qos-specs/{qos_id}	Shows details for a QoS specification.
PUT	/v2/{tenant_id}/qos-specs/{qos_id}	Sets keys in a QoS specification.
DELETE	/v2/{tenant_id}/qos-specs/{qos_id}	Deletes a QoS specification.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/associate	Associates a QoS specification with a volume type.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate	Disassociates a QoS specification from a volume type.

Method	URI	Description
GET	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate_all	Disassociates a QoS specification from all associations.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/associations	Lists all associations for a QoS specification.

1.11.1. Create QoS specification

Method	URI	Description
POST	/v2/{tenant_id}/qos-specs	Creates a QoS specification.

Specify one or more key and value pairs in the request body.

Normal response codes: 202

1.11.1.1. Request

This table shows the URI parameters for the create qos specification request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.69. Create QoS specification: JSON request

```
{
  "qos_specs": {
    "availability": "100",
    "name": "reliability-spec",
    "numberOfFailures": "0"
  }
}
```

This table shows the body parameters for the create qos specification request:

Name	Type	Description
qos_specs	Dict (Required)	A qos_specs object.
name	String (Required)	The name of the QoS specification.
consumer	String (Optional)	The consumer type.

Example 1.70. Create QoS specification: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<qos_specs name="performance-spec" delay="0" throughput="100" />
```

This operation does not accept a request body.

1.11.1.2. Response

Example 1.71. Create QoS specification: JSON response

```
{
  "qos_specs": {
    "specs": {
      "numberOfFailures": "0",
```

```

        "availability": "100"
      },
      "consumer": "back-end",
      "name": "reliability-spec",
      "id": "599ef437-1c99-42ec-9fc6-239d0519fef1"
    },
    "links": [
      {
        "href": "http://23.253.248.171:8776/
v2/bab7d5c60cd041a0a36f7c4b6e1dd978/qos_specs/
599ef437-1c99-42ec-9fc6-239d0519fef1",
        "rel": "self"
      },
      {
        "href": "http://23.253.248.171:8776/
bab7d5c60cd041a0a36f7c4b6e1dd978/qos_specs/
599ef437-1c99-42ec-9fc6-239d0519fef1",
        "rel": "bookmark"
      }
    ]
  }
}

```

This table shows the body parameters for the create qos specification response:

Name	Type	Description
qos_specs	Dict (Required)	A qos_specs object.
specs	Dict (Required)	A specs object.
consumer	String (Required)	The consumer type.
name	String (Required)	The name of the QoS specification.
id	UUID (Required)	The generated ID for the QoS specification.
links	List (Required)	The QoS specification links.

Example 1.72. Create QoS specification: XML response

```

<?xml version="1.0" encoding="UTF-8"?>
<qos_specs>
  <qos_spec consumer="back-end" id="e1f84cf3-6b1e-4f25-8130-3244f69ec7c1"
  name="performance-spec">
    <specs>
      <delay>0</delay>
      <throughput>100</throughput>
    </specs>
  </qos_spec>
</qos_specs>

```

This operation does not return a response body.

1.11.2. List QoS specs

Method	URI	Description
GET	/v2/{tenant_id}/qos-specs{?sort_key,sort_dir,limit,marker}	Lists quality of service (QoS) specifications.

Normal response codes: 200300

1.11.2.1. Request

This table shows the URI parameters for the list qos specs request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list qos specs request:

Name	Type	Description
sort_key	String (Optional)	Sorts by an image attribute. A valid value is name, status, container_format, disk_format, size, id, created_at, or updated_at. Default is created_at. The API uses the natural sorting direction of the sort_key attribute value.
sort_dir	String (Optional)	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is desc.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

1.11.2.2. Response

Example 1.73. List QoS specs: JSON response

```
{
  "qos_specs": [
    {
      "specs": {
        "availability": "100",
        "numberOfFailures": "0"
      },
      "consumer": "back-end",
      "name": "reliability-spec",
      "id": "0388d6c6-d5d4-42a3-b289-95205c50dd15"
    },
    {
      "specs": {
        "delay": "0",
        "throughput": "100"
      }
    }
  ]
}
```

```

    },
    "consumer": "back-end",
    "name": "performance-spec",
    "id": "ecfc6e2e-7117-44a4-8eec-f84d04f531a8"
  }
]
}

```

This table shows the body parameters for the list qos specs response:

Name	Type	Description
qos_specs	Dict (Required)	A qos_specs object.
specs	Dict (Required)	Specification key and value pairs.
consumer	String (Required)	The consumer type.
name	String (Required)	The name of the QoS specification.
id	UUID (Required)	The generated ID for the QoS specification.

Example 1.74. List QoS specs: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<qos_specs>
  <qos_spec consumer="back-end"
    id="0388d6c6-d5d4-42a3-b289-95205c50dd15"
    name="reliability-spec">
    <specs>
      <availability>100</availability>
      <numberOfFailures>0</numberOfFailures>
    </specs>
  </qos_spec>
  <qos_spec consumer="back-end"
    id="ecfc6e2e-7117-44a4-8eec-f84d04f531a8"
    name="performance-spec">
    <specs>
      <delay>0</delay>
      <throughput>100</throughput>
    </specs>
  </qos_spec>
</qos_specs>

```

This operation does not return a response body.

1.11.3. Show QoS specification details

Method	URI	Description
GET	/v2/{tenant_id}/qos-specs/{qos_id}	Shows details for a QoS specification.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

1.11.3.1. Request

This table shows the URI parameters for the show qos specification details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{qos_id}	String	The ID of the QoS specification.

This operation does not accept a request body.

1.11.3.2. Response

Example 1.75. Show QoS specification details: JSON response

```
{
  "qos_specs": {
    "specs": {
      "availability": "100",
      "numberOfFailures": "0"
    },
    "consumer": "back-end",
    "name": "reliability-spec",
    "id": "0388d6c6-d5d4-42a3-b289-95205c50dd15"
  },
  "links": [
    {
      "href": "http://23.253.228.211:8776/v2/e1cf63117ae74309a5bcc2002a23be8b/qos_specs/0388d6c6-d5d4-42a3-b289-95205c50dd15",
      "rel": "self"
    },
    {
      "href": "http://23.253.228.211:8776/v2/e1cf63117ae74309a5bcc2002a23be8b/qos_specs/0388d6c6-d5d4-42a3-b289-95205c50dd15",
      "rel": "bookmark"
    }
  ]
}
```

This table shows the body parameters for the show qos specification details response:

Name	Type	Description
qos_specs	Dict	A qos_specs object.

Name	Type	Description
	(Required)	
specs	Dict (Required)	Specification key and value pairs.
consumer	String (Required)	The consumer type.
name	String (Required)	The name of the QoS specification.
id	UUID (Required)	The generated ID for the QoS specification.
links	List (Required)	The QoS specification links.

Example 1.76. Show QoS specification details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<qos_specs>
  <qos_spec consumer="back-end"
    id="0388d6c6-d5d4-42a3-b289-95205c50dd15"
    name="reliability-spec">
    <specs>
      <availability>100</availability>
      <numberOfFailures>0</numberOfFailures>
    </specs>
  </qos_spec>
</qos_specs>
```

This operation does not return a response body.

1.11.4. Set keys in QoS specification

Method	URI	Description
PUT	/v2/{tenant_id}/qos-specs/{qos_id}	Sets keys in a QoS specification.

Normal response codes: 200

1.11.4.1. Request

This table shows the URI parameters for the set keys in qos specification request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{qos_id}	String	The ID of the QoS specification.

Example 1.77. Set keys in QoS specification: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<qos_specs delay="2" throughput="100"/>
```

Example 1.78. Set keys in QoS specification: JSON request

```
{
  "qos_specs": {
    "delay": "1"
  }
}
```

This operation does not accept a request body.

1.11.4.2. Response

Example 1.79. Set keys in QoS specification: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<qos_specs>
  <qos_spec/>
</qos_specs>
```

Example 1.80. Set keys in QoS specification: JSON response

```
{
  "qos_specs": {
    "delay": "1"
  }
}
```

This operation does not return a response body.

1.11.5. Delete QoS specification

Method	URI	Description
DELETE	/v2/{tenant_id}/qos-specs/{qos_id}	Deletes a QoS specification.

Normal response codes: 202

1.11.5.1. Request

This table shows the URI parameters for the delete qos specification request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{qos_id}	String	The ID of the QoS specification.
{force}	Boolean	To delete a QoS specification even if it is in-use, set to <code>true</code> . Default is <code>false</code> .

This operation does not accept a request body.

1.11.6. Associate QoS specification with volume type

Method	URI	Description
GET	/v2/{tenant_id}/qos-specs/{qos_id}/associate	Associates a QoS specification with a volume type.

Normal response codes: 202

1.11.6.1. Request

This table shows the URI parameters for the associate qos specification with volume type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{qos_id}	String	The ID of the QoS specification.
{volume_type_id}	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

1.11.7. Disassociate QoS specification from volume type

Method	URI	Description
GET	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate	Disassociates a QoS specification from a volume type.

Normal response codes: 202

1.11.7.1. Request

This table shows the URI parameters for the disassociate qos specification from volume type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{qos_id}	String	The ID of the QoS specification.
{volume_type_id}	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

1.11.8. Disassociate QoS specification from all associations

Method	URI	Description
GET	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate_all	Disassociates a QoS specification from all associations.

Normal response codes: 202

1.11.8.1. Request

This table shows the URI parameters for the disassociate qos specification from all associations request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{qos_id}	String	The ID of the QoS specification.

This operation does not accept a request body.

1.11.9. Get all associations for QoS specification

Method	URI	Description
GET	/v2/{tenant_id}/qos-specs/{qos_id}/associations	Lists all associations for a QoS specification.

Normal response codes: 200

1.11.9.1. Request

This table shows the URI parameters for the get all associations for qos specification request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{qos_id}	String	The ID of the QoS specification.

This operation does not accept a request body.

1.11.9.2. Response

Example 1.81. Get all associations for QoS specification: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<qos_specs>
  <qos_spec consumer="back-end"
    id="0388d6c6-d5d4-42a3-b289-95205c50dd15"
    name="reliability-spec">
    <specs>
      <availability>100</availability>
      <numberOfFailures>0</numberOfFailures>
    </specs>
  </qos_spec>
</qos_specs>
```

Example 1.82. Get all associations for QoS specification: JSON response

```
{
  "qos_specs": {
    "specs": {
      "availability": "100",
      "numberOfFailures": "0"
    },
    "consumer": "back-end",
    "name": "reliability-spec",
    "id": "0388d6c6-d5d4-42a3-b289-95205c50dd15"
  },
  "links": [
    {
      "href": "http://23.253.228.211:8776/v2/
e1cf63117ae74309a5bcc2002a23be8b/qos_specs/0388d6c6-d5d4-42a3-
b289-95205c50dd15",
      "rel": "self"
    }
  ]
}
```

```
{
  "href": "http://23.253.228.211:8776/
e1cf63117ae74309a5bcc2002a23be8b/qos_specs/0388d6c6-d5d4-42a3-
b289-95205c50dd15",
  "rel": "bookmark"
}
]
```

This operation does not return a response body.

1.12. Volume types (types)

Method	URI	Description
GET	/v2/{tenant_id}/types{?sort_key,sort_dir,limit,marker}	Lists volume types.
POST	/v2/{tenant_id}/types	Creates a volume type.
PUT	/v2/{tenant_id}/types/{volume_type_id}	Updates a volume type.
PUT	/v2/{tenant_id}/types/{volume_type_id}	Updates the extra specifications that are assigned to a volume type.
GET	/v2/{tenant_id}/types/{volume_type_id}	Shows details for a volume type.
DELETE	/v2/{tenant_id}/types/{volume_type_id}	Deletes a volume type.

1.12.1. List volume types

Method	URI	Description
GET	/v2/{tenant_id}/types{?sort_key,sort_dir,limit,marker}	Lists volume types.

Normal response codes: 200

1.12.1.1. Request

This table shows the URI parameters for the list volume types request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list volume types request:

Name	Type	Description
sort_key	String (Optional)	Sorts by an attribute. A valid value is name, status, container_format, disk_format, size, id, created_at, or updated_at. Default is created_at. The API uses the natural sorting direction of the sort_key attribute value.
sort_dir	String (Optional)	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is desc.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

1.12.1.2. Response

Example 1.83. List volume types: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_types
  xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content">
  <volume_type id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="SSD">
    <extra_specs>
      <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
  </volume_type>
  <volume_type id="8eb69a46-df97-4e41-9586-9a40a7533803" name="SATA">
  </volume_type>
</volume_types>
```

Example 1.84. List volume types: JSON response

```
{
```

```
    "volume_types": [  
      {  
        "extra_specs": {  
          "capabilities": "gpu"  
        },  
        "id": "6685584b-1eac-4da6-b5c3-555430cf68ff",  
        "name": "SSD"  
      },  
      {  
        "extra_specs": {},  
        "id": "8eb69a46-df97-4e41-9586-9a40a7533803",  
        "name": "SATA"  
      }  
    ]  
  }  
}
```

This operation does not return a response body.

1.12.2. Create volume type

Method	URI	Description
POST	/v2/{tenant_id}/types	Creates a volume type.

To create an environment with multiple-storage back ends, you must specify a volume type. Block Storage volume back ends are spawned as children to `cinder-volume`, and they are keyed from a unique queue. They are named `cinder-volume.HOST.BACKEND`. For example, `cinder-volume.ubuntu.lvmdriver`. When a volume is created, the scheduler chooses an appropriate back end to handle the request based on the volume type.

For information about how to use volume types to create multiple-storage back ends, see [Configure multiple-storage back ends](#).

Normal response codes: 200

1.12.2.1. Request

This table shows the URI parameters for the create volume type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.85. Create volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1" xmlns:os-volume-
type-access="http://docs.openstack.org/openstack-block-storage/2.0/ext/os-
volume-type-access/api/v2.0" name="vol-type-001" description="volume type
0001" os-volume-type-access:is_public="true">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 1.86. Create volume type: JSON request

```
{
  "volume_type": {
    "name": "vol-type-001",
    "description": "volume type 0001",
    "os-volume-type-access:is_public": true,
    "extra_specs": {
      "capabilities": "gpu"
    }
  }
}
```

This operation does not accept a request body.

1.12.2.2. Response

Example 1.87. Create volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<volume_type
  xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
  id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="vol-type-001"
  description="volume type 001" is_public="true">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 1.88. Create volume type: JSON response

```
{
  "volume_type": {
    "id": "6685584b-1eac-4da6-b5c3-555430cf68ff",
    "name": "vol-type-001",
    "description": "volume type 001",
    "is_public": "true",
    "extra_specs": {
      "capabilities": "gpu"
    }
  }
}
```

This operation does not return a response body.

1.12.3. Update volume type

Method	URI	Description
PUT	/v2/{tenant_id}/types/{volume_type_id}	Updates a volume type.

To create an environment with multiple-storage back ends, you must specify a volume type. The API spawns Block Storage volume back ends as children to `cinder-volume`, and keys them from a unique queue. The API names the back ends `cinder-volume.HOST.BACKEND`. For example, `cinder-volume.ubuntu.lvmdriver`. When you create a volume, the scheduler chooses an appropriate back end for the volume type to handle the request.

For information about how to use volume types to create multiple-storage back ends, see [Configure multiple-storage back ends](#).

Normal response codes: 200

1.12.3.1. Request

This table shows the URI parameters for the update volume type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_type_id}	UUID	The UUID for an existing volume type.

Example 1.89. Update volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1" xmlns:os-volume-type-access="http://docs.openstack.org/openstack-block-storage/2.0/ext/os-volume-type-access/api/v2.0" name="vol-type-001" description="volume type 0001" is_public="true">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 1.90. Update volume type: JSON request

```
{
  "volume_type": {
    "name": "vol-type-001",
    "description": "volume type 0001",
    "is_public": true,
    "extra_specs": {
      "capabilities": "gpu"
    }
  }
}
```

This operation does not accept a request body.

1.12.3.2. Response

Example 1.91. Update volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type
  xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
  id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="vol-type-001"
  description="volume type 001" is_public="true">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 1.92. Update volume type: JSON response

```
{
  "volume_type": {
    "id": "6685584b-1eac-4da6-b5c3-555430cf68ff",
    "name": "vol-type-001",
    "description": "volume type 001",
    "is_public": "true",
    "extra_specs": {
      "capabilities": "gpu"
    }
  }
}
```

This operation does not return a response body.

1.12.4. Update extra specs for a volume type

Method	URI	Description
PUT	/v2/{tenant_id}/types/{volume_type_id}	Updates the extra specifications that are assigned to a volume type.

Normal response codes: 200

1.12.4.1. Request

This table shows the URI parameters for the update extra specs for a volume type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_type_id}	UUID	The UUID for an existing volume type.

Example 1.93. Update extra specs for a volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1" xmlns:os-volume-
type-access="http://docs.openstack.org/openstack-block-storage/2.0/ext/os-
volume-type-access/api/v2.0" name="vol-type-001" description="volume type
0001" is_public="true">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 1.94. Update extra specs for a volume type: JSON request

```
{
  "volume_type": {
    "name": "vol-type-001",
    "description": "volume type 0001",
    "is_public": true,
    "extra_specs": {
      "capabilities": "gpu"
    }
  }
}
```

This operation does not accept a request body.

1.12.4.2. Response

Example 1.95. Update extra specs for a volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type
  xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
  id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="vol-type-001"
  description="volume type 001" is_public="true">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
```

```
</extra_specs>  
</volume_type>
```

Example 1.96. Update extra specs for a volume type: JSON response

```
{  
  "volume_type": {  
    "id": "6685584b-1eac-4da6-b5c3-555430cf68ff",  
    "name": "vol-type-001",  
    "description": "volume type 001",  
    "is_public": "true",  
    "extra_specs": {  
      "capabilities": "gpu"  
    }  
  }  
}
```

This operation does not return a response body.

1.12.5. Show volume type details

Method	URI	Description
GET	/v2/{tenant_id}/types/{volume_type_id}	Shows details for a volume type.

Normal response codes: 200

1.12.5.1. Request

This table shows the URI parameters for the show volume type details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_type_id}	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

1.12.5.2. Response

Example 1.97. Show volume type details: JSON response

```
{
  "volume_type": {
    "id": "6685584b-1eac-4da6-b5c3-555430cf68ff",
    "name": "vol-type-001",
    "description": "volume type 001",
    "is_public": "true",
    "extra_specs": {
      "capabilities": "gpu"
    }
  }
}
```

This table shows the body parameters for the show volume type details response:

Name	Type	Description
name	String (Required)	The name of the volume type.
extra_specs	Dict (Required)	A set of key and value pairs that contains the specifications for a volume type.

Example 1.98. Show volume type details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type
  xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
  id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="vol-type-001"
  description="volume type 001" is_public="true">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

```
</volume_type>
```

This operation does not return a response body.

1.12.6. Delete volume type

Method	URI	Description
DELETE	/v2/{tenant_id}/types/{volume_type_id}	Deletes a volume type.

Normal response codes: 202

1.12.6.1. Request

This table shows the URI parameters for the delete volume type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_type_id}	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

1.13. Volume snapshots (snapshots)

A snapshot is a point-in-time copy of the data that a volume contains.

When you create, list, or delete snapshots, these status values are possible:

Table 1.3. Snapshot statuses

Status	Description
creating	The snapshot is being created.
available	The snapshot is ready to use.
deleting	The snapshot is being deleted.
error	A snapshot creation error occurred.
error_deleting	A snapshot deletion error occurred.

Method	URI	Description
POST	/v2/{tenant_id}/snapshots	Creates a volume snapshot, which is a point-in-time, complete copy of a volume. You can create a volume from a snapshot.
GET	/v2/{tenant_id}/snapshots{?sort_key,sort_dir,limit,marker}	Lists all Block Storage snapshots, with summary information, that the tenant can access.
GET	/v2/{tenant_id}/snapshots/detail	Lists all Block Storage snapshots, with details, that the tenant can access.
GET	/v2/{tenant_id}/snapshots/{snapshot_id}	Shows details for a snapshot.
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}	Updates a snapshot.
DELETE	/v2/{tenant_id}/snapshots/{snapshot_id}	Deletes a snapshot.
GET	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Shows metadata for a snapshot.

Method	URI	Description
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Updates metadata for a snapshot.

1.13.1. Create snapshot

Method	URI	Description
POST	/v2/{tenant_id}/snapshots	Creates a volume snapshot, which is a point-in-time, complete copy of a volume. You can create a volume from a snapshot.

Normal response codes: 202

1.13.1.1. Request

This table shows the URI parameters for the create snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the body parameters for the create snapshot request:

Name	Type	Description
snapshot	String (Required)	A partial representation of a snapshot used in the creation process.
name	String (Optional)	The name of the snapshot. Default is None.
description	String (Optional)	A description for the snapshot. Default is None.
volume_id	UUID (Required)	To create a snapshot from an existing volume, specify the UUID of the existing volume.
force	Boolean (Optional)	Indicates whether to snapshot, even if the volume is attached. Default is false.

Example 1.99. Create snapshot: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot
  xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
  name="snap-001" description="Daily backup"
  volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" force="true"/>
```

Example 1.100. Create snapshot: JSON request

```
{
  "snapshot": {
    "name": "snap-001",
    "description": "Daily backup",
    "volume_id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
    "force": true
  }
}
```

This operation does not accept a request body.

1.13.1.2. Response

Example 1.101. Create snapshot: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshot status="creating" description="Daily backup"
  created_at="2013-02-25T03:56:53.081642"
  volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
  id="ffa9bc5e-1172-4021-acaf-cdc78a9584d" name="snap-001">
  <metadata/>
</snapshot>
```

Example 1.102. Create snapshot: JSON response

```
{
  "snapshot": {
    "status": "creating",
    "description": "Daily backup",
    "created_at": "2013-02-25T03:56:53.081642",
    "metadata": {},
    "volume_id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
    "size": 1,
    "id": "ffa9bc5e-1172-4021-acaf-cdc78a9584d",
    "name": "snap-001"
  }
}
```

This operation does not return a response body.

1.13.2. List snapshots

Method	URI	Description
GET	/v2/{tenant_id}/snapshots{?sort_key,sort_dir,limit,marker}	Lists all Block Storage snapshots, with summary information, that the tenant can access.

Normal response codes: 200

1.13.2.1. Request

This table shows the URI parameters for the list snapshots request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list snapshots request:

Name	Type	Description
sort_key	String (Optional)	Sorts by an attribute. A valid value is name, status, container_format, disk_format, size, id, created_at, or updated_at. Default is created_at. The API uses the natural sort-ing direction of the sort_key attribute value.
sort_dir	String (Optional)	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is desc.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker pa-rameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited re-quest.

This operation does not accept a request body.

1.13.2.2. Response

Example 1.103. List snapshots: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshots>
  <snapshot status="available"
    description="volume snapshot"
    created_at="2015-11-29 02:25:51+00:00"
    volume_id="173f7b48-c4c1-4e70-9acc-086b39073506"
    size="1" id="b1323cda-8e4b-41c1-afc5-2fc791809c8c"
    name="test-volume-snapshot">
    <metadata>
      <meta key="name">test</meta>
    </metadata>
  </snapshot>
</snapshots>
```

Example 1.104. List snapshots: JSON response

```
{
  "snapshots": [
    {
      "status": "available",
      "metadata": {
        "name": "test"
      },
      "name": "test-volume-snapshot",
      "volume_id": "173f7b48-c4c1-4e70-9acc-086b39073506",
      "created_at": "2015-11-29T02:25:51.000000",
      "size": 1,
      "id": "b1323cda-8e4b-41c1-afc5-2fc791809c8c",
      "description": "volume snapshot"
    }
  ]
}
```

This operation does not return a response body.

1.13.3. List snapshots with details

Method	URI	Description
GET	/v2/{tenant_id}/snapshots/detail	Lists all Block Storage snapshots, with details, that the tenant can access.

Normal response codes: 200

1.13.3.1. Request

This table shows the URI parameters for the list snapshots with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.13.3.2. Response

Example 1.105. List snapshots with details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshots
  xmlns:os-extended-snapshot-attributes="http://docs.openstack.org/volume/
ext/extended_snapshot_attributes/api/v1">
  <snapshot status="available"
    description="volume snapshot"
    created_at="2015-11-29 02:25:51+00:00"
    volume_id="173f7b48-c4c1-4e70-9acc-086b39073506"
    size="1" id="b1323cda-8e4b-41c1-afc5-2fc791809c8c"
    name="test-volume-snapshot"
    os-extended-snapshot-attributes:project_id=
"bab7d5c60cd041a0a36f7c4b6e1dd978"
    os-extended-snapshot-attributes:progress="100%">
    <metadata>
      <meta key="name">test</meta>
    </metadata>
  </snapshot>
</snapshots>
```

Example 1.106. List snapshots with details: JSON response

```
{
  "snapshots": [
    {
      "status": "available",
      "metadata": {
        "name": "test"
      },
      "os-extended-snapshot-attributes:progress": "100%",
      "name": "test-volume-snapshot",
      "volume_id": "173f7b48-c4c1-4e70-9acc-086b39073506",
      "os-extended-snapshot-attributes:project_id":
"bab7d5c60cd041a0a36f7c4b6e1dd978",
      "created_at": "2015-11-29T02:25:51.000000",
```

```
        "size": 1,  
        "id": "b1323cda-8e4b-41c1-afc5-2fc791809c8c",  
        "description": "volume snapshot"  
    }  
]  
}
```

This operation does not return a response body.

1.13.4. Show snapshot details

Method	URI	Description
GET	/v2/{tenant_id}/snapshots/{snapshot_id}	Shows details for a snapshot.

Normal response codes: 200

1.13.4.1. Request

This table shows the URI parameters for the show snapshot details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

1.13.4.2. Response

Example 1.107. Show snapshot details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshot
  xmlns:os-extended-snapshot-attributes="http://docs.openstack.org/openstack-
block-storage/2.0/content/Extended_Snapshot_Attributes.html"
  status="available" description="Very important"
  created_at="2013-02-25 04:13:17"
  volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
  id="2bb856e1-b3d8-4432-a858-09e4ce939389" name="snap-001"
  os-extended-snapshot-attributes:project_id=
"0c2eba2c5af04d3f9e9d0d410b371fde"
  os-extended-snapshot-attributes:progress="100%">
  <metadata/>
</snapshot>
```

Example 1.108. Show snapshot details: JSON response

```
{
  "snapshot": {
    "status": "available",
    "os-extended-snapshot-attributes:progress": "100%",
    "description": "Daily backup",
    "created_at": "2013-02-25T04:13:17.000000",
    "metadata": {},
    "volume_id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
    "os-extended-snapshot-attributes:project_id":
"0c2eba2c5af04d3f9e9d0d410b371fde",
    "size": 1,
    "id": "2bb856e1-b3d8-4432-a858-09e4ce939389",
    "name": "snap-001"
  }
}
```

This operation does not return a response body.

1.13.5. Update snapshot

Method	URI	Description
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}	Updates a snapshot.

Normal response codes: 200

1.13.5.1. Request

This table shows the URI parameters for the update snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

Example 1.109. Update snapshot: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot
  xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
  name="snap-002" description="This is yet, another snapshot."/>
```

Example 1.110. Update snapshot: JSON request

```
{
  "snapshot": {
    "name": "snap-002",
    "description": "This is yet, another snapshot."
  }
}
```

This operation does not accept a request body.

1.13.5.2. Response

Example 1.111. Update snapshot: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshot
  xmlns:os-extended-snapshot-attributes="http://docs.openstack-
block-storage/2.0/content/Extended_Snapshot_Attributes.html"
  status="available"
  description="This is yet, another snapshot"
  created_at="2013-02-20T08:11:34.000000"
  volume_id="2402b902-0b7a-458c-9c07-7435a826f794"
  size="1"
  id="4b502fcb-1f26-45f8-9fe5-3b9a0a52eaf2"
  name="snap-002"
  os-extended-snapshot-attributes:project_id=
"0c2eba2c5af04d3f9e9d0d410b371fde"
  os-extended-snapshot-attributes:progress="100%">
  <metadata/>
</snapshot>
```

Example 1.112. Update snapshot: JSON response

```
{
  "snapshot": {
    "created_at": "2013-02-20T08:11:34.000000",
    "description": "This is yet, another snapshot",
    "name": "snap-002",
    "id": "4b502fcb-1f26-45f8-9fe5-3b9a0a52eaf2",
    "size": 1,
    "status": "available",
    "volume_id": "2402b902-0b7a-458c-9c07-7435a826f794"
  }
}
```

This operation does not return a response body.

1.13.6. Delete snapshot

Method	URI	Description
DELETE	/v2/{tenant_id}/snapshots/{snapshot_id}	Deletes a snapshot.

Normal response codes: 202

1.13.6.1. Request

This table shows the URI parameters for the delete snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

1.13.7. Show snapshot metadata

Method	URI	Description
GET	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Shows metadata for a snapshot.

Normal response codes: 200

1.13.7.1. Request

This table shows the URI parameters for the show snapshot metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

1.13.7.2. Response

Example 1.113. Show snapshot metadata: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
  <meta key="name">test</meta>
</metadata>
```

Example 1.114. Show snapshot metadata: JSON response

```
{
  "metadata": {
    "name": "test"
  }
}
```

This operation does not return a response body.

1.13.8. Update snapshot metadata

Method	URI	Description
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Updates metadata for a snapshot.

Replaces metadata items that match keys. Does not modify items that are not in the request.

Normal response codes: 200

1.13.8.1. Request

This table shows the URI parameters for the update snapshot metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

Example 1.115. Update snapshot metadata: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata>
  <meta key="key">v2</meta>
</metadata>
```

Example 1.116. Update snapshot metadata: JSON request

```
{
  "metadata": {
    "key": "v2"
  }
}
```

This operation does not accept a request body.

1.13.8.2. Response

Example 1.117. Update snapshot metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
  <meta key="key">v2</meta>
</metadata>
```

Example 1.118. Update snapshot metadata: JSON response

```
{
  "metadata": {
    "key": "v2"
  }
}
```

This operation does not return a response body.

1.14. Volume manage extension (os-volume-manage)

Creates volumes by using existing storage instead of allocating new storage.

Method	URI	Description
POST	/v2/{tenant_id}/os-volume-manage	Creates a Block Storage volume by using existing storage rather than allocating new storage.

1.14.1. Manage existing volume

Method	URI	Description
POST	/v2/{tenant_id}/os-volume-manage	Creates a Block Storage volume by using existing storage rather than allocating new storage.

The caller must specify a reference to an existing storage volume in the *ref* parameter in the request. Although each storage driver might interpret this reference differently, the driver should accept a reference structure that contains either a *source-volume-id* or *source-volume-name* element, if possible.

The API chooses the size of the volume by rounding up the size of the existing storage volume to the next gibibyte (GiB).

Normal response codes: 202

1.14.1.1. Request

This table shows the URI parameters for the manage existing volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.119. Show manage existing request: JSON

```
{
  "volume": {
    "host": "geraint-VirtualBox",
    "ref": {
      "source-volume-name": "existingLV",
      "source-volume-id": "1234"
    },
    "name": "New Volume",
    "availability_zone": "az2",
    "description": "Volume imported from existingLV",
    "volume_type": null,
    "bootable": true,
    "metadata": {
      "key1": "value1",
      "key2": "value2"
    }
  }
}
```

1.14.1.2. Response

Example 1.120. Manage existing volume: JSON response

The response is the same as the volume create response.

```
{
  "volume": {
    "status": "creating",
    "user_id": "eae1472b5fc5496998a3d06550929e7e",
  }
}
```

```

    "attachments": [],
    "links": [
      {
        "href": "http://10.0.2.15:8776/v2/87c8522052ca4eed98bc672b4c1a3ddb/volumes/23cf872b-c781-4cd4-847d-5f2ec8cbd91c",
        "rel": "self"
      },
      {
        "href": "http://10.0.2.15:8776/v2/87c8522052ca4eed98bc672b4c1a3ddb/volumes/23cf872b-c781-4cd4-847d-5f2ec8cbd91c",
        "rel": "bookmark"
      }
    ],
    "availability_zone": "az2",
    "bootable": "false",
    "encrypted": "false",
    "created_at": "2014-07-18T00:12:54.000000",
    "description": "Volume imported from existingLV",
    "os-vol-tenant-attr:tenant_id": "87c8522052ca4eed98bc672b4c1a3ddb",
    "volume_type": null,
    "name": "New Volume",
    "source_volid": null,
    "snapshot_id": null,
    "metadata": {
      "key2": "value2",
      "key1": "value1"
    },
    "id": "23cf872b-c781-4cd4-847d-5f2ec8cbd91c",
    "size": 0
  }
}

```

1.15. Volume image metadata extension (os-vol-image-meta)

Shows image metadata that is associated with a volume.

Method	URI	Description
GET	/v2/{tenant_id}/os-vol-image-meta	Shows image metadata for a volume.

1.15.1. Show image metadata for volume

Method	URI	Description
GET	/v2/{tenant_id}/os-vol-image-meta	Shows image metadata for a volume.

When the request is made, the caller must specify a reference to an existing storage volume in the `ref` element. Each storage driver may interpret the existing storage volume reference differently but should accept a reference structure containing either a `source-volume-id` or `source-volume-name` element, if possible.

Normal response codes: 202

1.15.1.1. Request

This table shows the URI parameters for the show image metadata for volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.121. Show manage existing request: JSON

```
{
  "volume": {
    "host": "geraint-VirtualBox",
    "ref": {
      "source-volume-name": "existingLV",
      "source-volume-id": "1234"
    },
    "name": "New Volume",
    "availability_zone": "az2",
    "description": "Volume imported from existingLV",
    "volume_type": null,
    "bootable": true,
    "metadata": {
      "key1": "value1",
      "key2": "value2"
    }
  }
}
```

1.15.1.2. Response

Example 1.122. Show image metadata for volume: JSON response

The response is the same as for a volume create API call. The size of the volume is chosen by inspecting the size of the existing storage volume and rounding up to the next gibibyte (GiB).

```
{
  "volume": {
    "status": "creating",
    "user_id": "eae1472b5fc5496998a3d06550929e7e",
    "attachments": [],
    "links": [
```

```

        {
            "href": "http://10.0.2.15:8776/v2/
87c8522052ca4eed98bc672b4c1a3ddb/volumes/23cf872b-
c781-4cd4-847d-5f2ec8cbd91c",
            "rel": "self"
        },
        {
            "href": "http://10.0.2.15:8776/
87c8522052ca4eed98bc672b4c1a3ddb/volumes/23cf872b-
c781-4cd4-847d-5f2ec8cbd91c",
            "rel": "bookmark"
        }
    ],
    "availability_zone": "az2",
    "bootable": "false",
    "encrypted": "false",
    "created_at": "2014-07-18T00:12:54.000000",
    "description": "Volume imported from existingLV",
    "os-vol-tenant-attr:tenant_id": "87c8522052ca4eed98bc672b4c1a3ddb",
    "volume_type": null,
    "name": "New Volume",
    "source_volid": null,
    "snapshot_id": null,
    "metadata": {
        "key2": "value2",
        "key1": "value1"
    },
    "id": "23cf872b-c781-4cd4-847d-5f2ec8cbd91c",
    "size": 0
}

```

1.16. Back-end storage pools

Administrator only. Lists all back-end storage pools that are known to the scheduler service.

Method	URI	Description
GET	/v2/{tenant_id}/scheduler-stats/ get_pools{?detail}	Lists all back-end storage pools.

1.16.1. List back-end storage pools

Method	URI	Description
GET	/v2/{tenant_id}/scheduler-stats/get_pools{?detail}	Lists all back-end storage pools.

Normal response codes: 200

1.16.1.1. Request

This table shows the URI parameters for the list back-end storage pools request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list back-end storage pools request:

Name	Type	Description
detail	Boolean (Optional)	Indicates whether to show pool details or only pool names in the response. Set to true to show pool details. Set to false to show only pool names. Default is false.

This operation does not accept a request body.

1.16.1.2. Response

Example 1.123. List back-end storage pools: JSON response

```
{
  "pools": [
    {
      "name": "pool1",
      "capabilities": {
        "updated": "2014-10-28T00:00:00-00:00",
        "total_capacity": 1024,
        "free_capacity": 100,
        "volume_backend_name": "pool1",
        "reserved_percentage": 0,
        "driver_version": "1.0.0",
        "storage_protocol": "iSCSI",
        "QoS_support": false
      }
    },
    {
      "name": "pool2",
      "capabilities": {
        "updated": "2014-10-28T00:00:00-00:00",
        "total_capacity": 512,
        "free_capacity": 200,
        "volume_backend_name": "pool2",
        "reserved_percentage": 0,
        "driver_version": "1.0.1",
        "storage_protocol": "iSER",

```

```
    "QoS_support": true
  }
]
}
```

1.17. Volume transfer

Transfers a volume from one user to another user.

Method	URI	Description
POST	/v2/{tenant_id}/os-volume-transfer	Creates a volume transfer.
GET	/v2/{tenant_id}/os-volume-transfer	Lists volume transfers.
GET	/v2/{tenant_id}/os-volume-transfer/detail	Lists volume transfers, with details.
GET	/v2/{tenant_id}/os-volume-transfer/{transfer_id}	Shows details for a volume transfer.
DELETE	/v2/{tenant_id}/os-volume-transfer/{transfer_id}	Deletes a volume transfer.
POST	/v2/{tenant_id}/os-volume-transfer/{transfer_id}/accept	Accepts a volume transfer.

1.17.1. Create volume transfer

Method	URI	Description
POST	/v2/{tenant_id}/os-volume-transfer	Creates a volume transfer.

Normal response codes: 202

1.17.1.1. Request

This table shows the URI parameters for the create volume transfer request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.124. Create volume transfer: JSON request

```
{
  "transfer": {
    "volume_id": "c86b9af4-151d-4ead-b62c-5fb967af0e37",
    "name": "first volume"
  }
}
```

1.17.1.2. Response

Example 1.125. Create volume transfer: JSON response

```
{
  "transfer": {
    "id": "1a7059f5-8ed7-45b7-8d05-2811e5d09f24",
    "created_at": "2015-02-25T03:56:53.081642",
    "name": "first volume",
    "volume_id": "c86b9af4-151d-4ead-b62c-5fb967af0e37",
    "auth_key": "9266c59563c84664",
    "links": [
      {
        "href": "http://localhost/v2/firstproject/volumes/3",
        "rel": "self"
      },
      {
        "href": "http://localhost/firstproject/volumes/3",
        "rel": "bookmark"
      }
    ]
  }
}
```

1.17.2. List volume transfers

Method	URI	Description
GET	/v2/{tenant_id}/os-volume-transfer	Lists volume transfers.

Normal response codes: 200

1.17.2.1. Request

This table shows the URI parameters for the list volume transfers request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.17.2.2. Response

Example 1.126. List volume transfers: JSON response

```
{
  "transfers": [
    {
      "id": "cac5c677-73a9-4288-bb9c-b2ebfb547377",
      "name": "first volume transfer",
      "volume_id": "894623a6-e901-4312-aa06-4275e6321cce",
      "links": [
        {
          "href": "http://localhost/v2/firstproject/volumes/1",
          "rel": "self"
        },
        {
          "href": "http://localhost/firstproject/volumes/1",
          "rel": "bookmark"
        }
      ]
    },
    {
      "id": "f26c0dee-d20d-4e80-8dee-a8d91b9742a1",
      "name": "second volume transfer",
      "volume_id": "673db275-379f-41af-8371-e1652132b4c1",
      "links": [
        {
          "href": "http://localhost/v2/firstproject/volumes/2",
          "rel": "self"
        },
        {
          "href": "http://localhost/firstproject/volumes/2",
          "rel": "bookmark"
        }
      ]
    }
  ]
}
```

1.17.3. List volume transfers, with details

Method	URI	Description
GET	/v2/{tenant_id}/os-volume-transfer/detail	Lists volume transfers, with details.

Normal response codes: 200

1.17.3.1. Request

This table shows the URI parameters for the list volume transfers, with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.17.3.2. Response

Example 1.127. List volume transfers, with details: JSON response

```
{
  "transfers": [
    {
      "id": "cac5c677-73a9-4288-bb9c-b2ebfb547377",
      "created_at": "2015-02-25T03:56:53.081642",
      "name": "first volume transfer",
      "volume_id": "894623a6-e901-4312-aa06-4275e6321cce",
      "links": [
        {
          "href": "http://localhost/v2/firstproject/volumes/1",
          "rel": "self"
        },
        {
          "href": "http://localhost/firstproject/volumes/1",
          "rel": "bookmark"
        }
      ]
    },
    {
      "id": "f26c0dee-d20d-4e80-8dee-a8d91b9742a1",
      "created_at": "2015-03-25T03:56:53.081642",
      "name": "second volume transfer",
      "volume_id": "673db275-379f-41af-8371-e1652132b4c1",
      "links": [
        {
          "href": "http://localhost/v2/firstproject/volumes/2",
          "rel": "self"
        },
        {
          "href": "http://localhost/firstproject/volumes/2",
          "rel": "bookmark"
        }
      ]
    }
  ]
}
```

```
    ]  
}
```

1.17.4. Show volume transfer details

Method	URI	Description
GET	/v2/{tenant_id}/os-volume-transfer/{transfer_id}	Shows details for a volume transfer.

Normal response codes: 200

1.17.4.1. Request

This table shows the URI parameters for the show volume transfer details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{transfer_id}	UUID	The unique identifier for a volume transfer.

This operation does not accept a request body.

1.17.4.2. Response

Example 1.128. Show volume transfer details: JSON response

```
{
  "transfer": {
    "id": "cac5c677-73a9-4288-bb9c-b2ebfb547377",
    "created_at": "2015-02-25T03:56:53.081642",
    "name": "first volume transfer",
    "volume_id": "894623a6-e901-4312-aa06-4275e6321cce",
    "links": [
      {
        "href": "http://localhost/v2/firstproject/volumes/1",
        "rel": "self"
      },
      {
        "href": "http://localhost/firstproject/volumes/1",
        "rel": "bookmark"
      }
    ]
  }
}
```

1.17.5. Delete volume transfer

Method	URI	Description
DELETE	/v2/{tenant_id}/os-volume-transfer/{transfer_id}	Deletes a volume transfer.

Normal response codes: 202

1.17.5.1. Request

This table shows the URI parameters for the delete volume transfer request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{transfer_id}	UUID	The unique identifier for a volume transfer.

This operation does not accept a request body.

1.17.6. Accept volume transfer

Method	URI	Description
POST	/v2/{tenant_id}/os-volume-transfer/{transfer_id}/accept	Accepts a volume transfer.

Normal response codes: 202

1.17.6.1. Request

This table shows the URI parameters for the accept volume transfer request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{transfer_id}	UUID	The unique identifier for a volume transfer.

Example 1.129. Accept volume transfer: JSON request

```
{
  "accept": {
    "auth_key": "9266c59563c84664"
  }
}
```

1.17.6.2. Response

Example 1.130. Accept volume transfer: JSON response

```
{
  "transfer": {
    "id": "cac5c677-73a9-4288-bb9c-b2ebfb547377",
    "name": "first volume transfer",
    "volume_id": "894623a6-e901-4312-aa06-4275e6321cce",
    "links": [
      {
        "href": "http://localhost/v2/firstproject/volumes/1",
        "rel": "self"
      },
      {
        "href": "http://localhost/firstproject/volumes/1",
        "rel": "bookmark"
      }
    ]
  }
}
```

1.18. Consistency groups

Consistency groups enable you to create snapshots at the exact same point in time from multiple volumes. For example, a database might place its tables, logs, and configuration on separate volumes. To restore this database from a previous point in time, it makes sense to restore the logs, tables, and configuration together from the exact same point in time.

Use the `policy.json` file to grant permissions for these actions to limit roles.

Method	URI	Description
GET	/v2/{tenant_id}/consistencygroups {?sort_key,sort_dir,limit,marker}	Lists consistency groups.
POST	/v2/{tenant_id}/consistencygroups	Creates a consistency group.
GET	/v2/{tenant_id}/consistency- groups/detail{?sort_key,sort_dir, limit,marker}	Lists consistency groups with details.
POST	/v2/{tenant_id}/consistency- groups/create_from_src	Creates a consistency group from source.
GET	/v2/{tenant_id}/consistency- groups/{consistencygroup_id}	Shows details for a consistency group.
POST	/v2/{tenant_id}/consistency- groups/{consistencygroup_id}/ delete	Deletes a consistency group.
PUT	/v2/{tenant_id}/consistency- groups/{consistencygroup_id}/up- date	Updates a consistency group.

1.18.1. List consistency groups

Method	URI	Description
GET	/v2/{tenant_id}/consistencygroups {?sort_key,sort_dir,limit,marker}	Lists consistency groups.

Normal response codes: 200

1.18.1.1. Request

This table shows the URI parameters for the list consistency groups request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list consistency groups request:

Name	Type	Description
sort_key	String (Optional)	Sorts by an image attribute. A valid value is name, status, container_format, disk_format, size, id, created_at, or updated_at. Default is created_at. The API uses the natural sorting direction of the sort_key attribute value.
sort_dir	String (Optional)	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is desc.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

1.18.1.2. Response

Example 1.131. List consistency groups: JSON response

```
{
  "consistencygroups": [
    {
      "id": "6f519a48-3183-46cf-a32f-41815f813986",
      "name": "my-cg1"
    },
    {
      "id": "aed36625-a6d7-4681-ba59-c7ba3d18c148",
      "name": "my-cg2"
    }
  ]
}
```

1.18.2. Create consistency group

Method	URI	Description
POST	/v2/{tenant_id}/consistencygroups	Creates a consistency group.

Normal response codes: 202

1.18.2.1. Request

This table shows the URI parameters for the create consistency group request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.132. Create consistency group: JSON request

```
{
  "consistencygroup": {
    "name": "firstcg",
    "description": "first consistency group",
    "volume_types": [
      "type1",
      "type2"
    ],
    "user_id": "6f519a48-3183-46cf-a32f-41815f814546",
    "project_id": "6f519a48-3183-46cf-a32f-41815f815555",
    "availability_zone": "az0",
    "status": "creating"
  }
}
```

1.18.2.2. Response

Example 1.133. Create consistency group: JSON response

```
{
  "consistencygroup": {
    "id": "6f519a48-3183-46cf-a32f-41815f816666",
    "name": "firstcg"
  }
}
```

1.18.3. List consistency groups with details

Method	URI	Description
GET	/v2/{tenant_id}/consistency-groups/detail{?sort_key,sort_dir,limit,marker}	Lists consistency groups with details.

Normal response codes: 200

1.18.3.1. Request

This table shows the URI parameters for the list consistency groups with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list consistency groups with details request:

Name	Type	Description
sort_key	String (Optional)	Sorts by an image attribute. A valid value is name, status, container_format, disk_format, size, id, created_at, or updated_at. Default is created_at. The API uses the natural sort-ing direction of the sort_key attribute value.
sort_dir	String (Optional)	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is desc.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker pa-rameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited re-quest.

This operation does not accept a request body.

1.18.3.2. Response

Example 1.134. List consistency groups with details: JSON response

```
{
  "consistencygroups": [
    {
      "id": "6f519a48-3183-46cf-a32f-41815f813986",
      "status": "available",
      "availability_zone": "az1",
      "created_at": "2015-09-16T09:28:52.000000",
      "name": "my-cg1",
      "description": "my first consistency group"
    },
    {
      "id": "aed36625-a6d7-4681-ba59-c7ba3d18c148",
      "status": "error",
      "availability_zone": "az2",
```

```
        "created_at": "2015-09-16T09:31:15.000000",  
        "name": "my-cg2",  
        "description": "Edited description"  
    }  
]  
}
```

1.18.4. Create consistency group from source

Method	URI	Description
POST	/v2/{tenant_id}/consistency-groups/create_from_src	Creates a consistency group from source.

Normal response codes: 202

1.18.4.1. Request

This table shows the URI parameters for the create consistency group from source request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.135. Create consistency group from source: JSON request

```
{
  "consistencygroup-from-src": {
    "name": "firstcg",
    "description": "first consistency group",
    "cgsnapshot_id": "6f519a48-3183-46cf-a32f-41815f813986",
    "source_cgid": "6f519a48-3183-46cf-a32f-41815f814546",
    "user_id": "6f519a48-3183-46cf-a32f-41815f815555",
    "project_id": "6f519a48-3183-46cf-a32f-41815f814444",
    "status": "creating"
  }
}
```

1.18.4.2. Response

Example 1.136. Create consistency group from source: JSON response

```
{
  "consistencygroup": {
    "id": "6f519a48-3183-46cf-a32f-41815f816666",
    "name": "firstcg"
  }
}
```

1.18.5. Show consistency group details

Method	URI	Description
GET	/v2/{tenant_id}/consistency-groups/{consistencygroup_id}	Shows details for a consistency group.

Normal response codes: 200

1.18.5.1. Request

This table shows the URI parameters for the show consistency group details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{consistencygroup_id}	String	The ID of the consistency group.

This operation does not accept a request body.

1.18.5.2. Response

Example 1.137. Show consistency group details: JSON response

```
{
  "consistencygroup": {
    "id": "6f519a48-3183-46cf-a32f-41815f813986",
    "status": "available",
    "availability_zone": "az1",
    "created_at": "2015-09-16T09:28:52.000000",
    "name": "my-cg1",
    "description": "my first consistency group"
  }
}
```


1.18.6. Delete consistency group

Method	URI	Description
POST	/v2/{tenant_id}/consistency-groups/{consistencygroup_id}/delete	Deletes a consistency group.

Normal response codes: 202

1.18.6.1. Request

This table shows the URI parameters for the delete consistency group request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{consistencygroup_id}	String	The ID of the consistency group.

Example 1.138. Delete consistency group: JSON request

```
{
  "consistencygroup": {
    "force": false
  }
}
```

1.18.7. Update consistency group

Method	URI	Description
PUT	/v2/{tenant_id}/consistency-groups/{consistencygroup_id}/update	Updates a consistency group.

Normal response codes: 202

1.18.7.1. Request

This table shows the URI parameters for the update consistency group request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{consistencygroup_id}	String	The ID of the consistency group.

This table shows the body parameters for the update consistency group request:

Name	Type	Description
name	String (Optional)	The consistency group name.
description	String (Optional)	The consistency group description.
add_volumes	String (Optional)	One or more volume UUIDs, separated by commas, to add to the volume consistency group.
remove_volumes	String (Optional)	One or more volume UUIDs, separated by commas, to remove from the volume consistency group.

Example 1.139. Update consistency group: XML request

```
{
  "consistencygroup": {
    "name": "my_cg",
    "description": "My consistency group",
    "add_volumes": "volume-uuid-1,volume-uuid-2",
    "remove_volumes": "volume-uuid-8,volume-uuid-9"
  }
}
```

This operation does not accept a request body.

1.19. Consistency group snapshots

Lists all, lists all with details, shows details for, creates, and deletes consistency group snapshots.

Method	URI	Description
GET	/v2/{tenant_id}/cgsnapshots	Lists all consistency group snapshots.

Method	URI	Description
POST	/v2/{tenant_id}/cgsnapshots	Creates a consistency group snapshot.
GET	/v2/{tenant_id}/cgsnapshots/detail	Lists all consistency group snapshots with details.
GET	/v2/{tenant_id}/cgsnapshots/{cgsnapshot_id}	Shows details for a consistency group snapshot.
DELETE	/v2/{tenant_id}/cgsnapshots/{cgsnapshot_id}	Deletes a consistency group snapshot.

1.19.1. List consistency group snapshots

Method	URI	Description
GET	/v2/{tenant_id}/cgsnapshots	Lists all consistency group snapshots.

Normal response codes: 200

1.19.1.1. Request

This table shows the URI parameters for the list consistency group snapshots request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.19.1.2. Response

Example 1.140. List consistency group snapshots: JSON response

```
{
  "cgsnapshots": [
    {
      "id": "6f519a48-3183-46cf-a32f-41815f813986",
      "name": "my-cg1"
    },
    {
      "id": "aed36625-a6d7-4681-ba59-c7ba3d18c148",
      "name": "my-cg2"
    }
  ]
}
```

1.19.2. Create consistency group snapshot

Method	URI	Description
POST	/v2/{tenant_id}/cgsnapshots	Creates a consistency group snapshot.

Normal response codes: 202

1.19.2.1. Request

This table shows the URI parameters for the create consistency group snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.141. Create consistency group snapshot: JSON request

```
{
  "cgsnapshot": {
    "consistencygroup_id": "6f519a48-3183-46cf-a32f-41815f814546",
    "name": "firstcg",
    "description": "first consistency group",
    "user_id": "6f519a48-3183-46cf-a32f-41815f814444",
    "project_id": "6f519a48-3183-46cf-a32f-41815f815555",
    "status": "creating"
  }
}
```

This operation does not accept a request body.

1.19.2.2. Response

Example 1.142. Create consistency group snapshot: JSON response

```
{
  "cgsnapshot": {
    "id": "6f519a48-3183-46cf-a32f-41815f816666",
    "name": "firstcg"
  }
}
```

This operation does not return a response body.

1.19.3. List consistency group snapshots with details

Method	URI	Description
GET	/v2/{tenant_id}/cgsnapshots/detail	Lists all consistency group snapshots with details.

Normal response codes: 200

1.19.3.1. Request

This table shows the URI parameters for the list consistency group snapshots with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.19.3.2. Response

Example 1.143. List consistency group snapshots with details: JSON response

```
{
  "cgsnapshots": [
    {
      "id": "6f519a48-3183-46cf-a32f-41815f813986",
      "consistencygroup_id": "6f519a48-3183-46cf-a32f-41815f814444",
      "status": "available",
      "created_at": "2015-09-16T09:28:52.000000",
      "name": "my-cg1",
      "description": "my first consistency group"
    },
    {
      "id": "aed36625-a6d7-4681-ba59-c7ba3d18c148",
      "consistencygroup_id": "aed36625-a6d7-4681-ba59-c7ba3d18dddd",
      "status": "error",
      "created_at": "2015-09-16T09:31:15.000000",
      "name": "my-cg2",
      "description": "Edited description"
    }
  ]
}
```

This operation does not return a response body.

1.19.4. Show consistency group snapshot details

Method	URI	Description
GET	/v2/{tenant_id}/cgsnapshots/{cgsnapshot_id}	Shows details for a consistency group snapshot.

Normal response codes: 200

1.19.4.1. Request

This table shows the URI parameters for the show consistency group snapshot details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{cgsnapshot_id}	String	The ID of the consistency group snapshot.

This operation does not accept a request body.

1.19.4.2. Response

Example 1.144. Show consistency group snapshot details: JSON response

```
{
  "cgsnapshot": {
    "id": "6f519a48-3183-46cf-a32f-41815f813986",
    "consistencygroup_id": "6f519a48-3183-46cf-a32f-41815f814444",
    "status": "available",
    "created_at": "2015-09-16T09:28:52.000000",
    "name": "my-cg1",
    "description": "my first consistency group"
  }
}
```

This operation does not return a response body.

1.19.5. Delete consistency group snapshot

Method	URI	Description
DELETE	/v2/{tenant_id}/cgsnapshots/{cgsnapshot_id}	Deletes a consistency group snapshot.

Normal response codes: 202

1.19.5.1. Request

This table shows the URI parameters for the delete consistency group snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{cgsnapshot_id}	String	The ID of the consistency group snapshot.

This operation does not accept a request body.

2. Block Storage API v1 (DEPRECATED)

Block Storage API v1 is deprecated in Kilo.

Manages volumes and snapshots for use with the Block Storage API, also known as cinder services.

When you create, list, or delete volumes, the following status values are possible:

- **CREATING.** The volume is being created.
- **AVAILABLE.** The volume is ready to attach to an instance.
- **ATTACHING.** The volume is attaching to an instance.
- **IN-USE.** The volume is attached to an instance.
- **DELETING.** The volume is being deleted.
- **ERROR.** An error occurred.
- **ERROR_DELETING.** A volume deletion error occurred.

Method	URI	Description
API versions		
GET	/	Lists information about all Block Storage API versions.
GET	/v1	Shows Block Storage API v1 details.
Volumes		
POST	/v1/{tenant_id}/volumes	Creates a volume.
GET	/v1/{tenant_id}/volumes	Lists all volumes.
GET	/v1/{tenant_id}/volumes/detail	Lists all volumes, with details.
GET	/v1/{tenant_id}/volumes/{volume_id}	Shows details for a volume.
DELETE	/v1/{tenant_id}/volumes/{volume_id}	Deletes a volume.
Volume types		
GET	/v1/{tenant_id}/types	Lists volume types.
POST	/v1/{tenant_id}/types	Creates a volume type.
PUT	/v1/{tenant_id}/types/{volume_type_id}	Updates a volume type.
PUT	/v1/{tenant_id}/types/{volume_type_id}	Updates the extra specifications for a volume type.
GET	/v1/{tenant_id}/types/{volume_type_id}	Shows details for a volume type.
DELETE	/v1/{tenant_id}/types/{volume_type_id}	Deletes a volume type.
Snapshots		
POST	/v1/{tenant_id}/snapshots	Creates a snapshot.
GET	/v1/{tenant_id}/snapshots	Lists all snapshots.
GET	/v1/{tenant_id}/snapshots/detail	Lists all snapshots, with details.
GET	/v1/{tenant_id}/snapshots/{snapshot_id}	Shows details for a snapshot.
DELETE	/v1/{tenant_id}/snapshots/{snapshot_id}	Deletes a snapshot.
GET	/v1/{tenant_id}/snapshots/{snapshot_id}/metadata	Shows metadata for a snapshot.

Method	URI	Description
PUT	/v1/{tenant_id}/snapshots/{snapshot_id}/metadata	Updates metadata for a snapshot.
Quota sets extension (os-quota-sets)		
GET	/v1/{tenant_id}/os-quota-sets/{tenant_id}?usage	Shows quotas for a tenant.
PUT	/v1/{tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.
DELETE	/v1/{tenant_id}/os-quota-sets/{tenant_id}	Deletes quotas for a tenant so the quotas revert to default values.
GET	/v1/{tenant_id}/os-quota-sets/default	Shows default quotas for a tenant.
GET	/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a tenant and user.
POST	/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a tenant and user.
DELETE	/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Deletes quotas for a user so that the quotas revert to default values.
GET	/v1/{tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}	Shows details for quotas for a tenant and user.

2.1. API versions

Lists information about API versions.

Method	URI	Description
GET	/	Lists information about all Block Storage API versions.
GET	/v1	Shows Block Storage API v1 details.

2.1.1. List API versions

Method	URI	Description
GET	/	Lists information about all Block Storage API versions.

Normal response codes: 200300

2.1.1.1. Request

This operation does not accept a request body.

2.1.1.2. Response

Example 2.1. List API versions: JSON response

```
{
  "versions": [
    {
      "id": "v1.0",
      "links": [
        {
          "href": "http://23.253.211.234:8776/v1/",
          "rel": "self"
        }
      ],
      "status": "DEPRECATED",
      "updated": "2014-06-28T12:20:21Z"
    },
    {
      "id": "v2.0",
      "links": [
        {
          "href": "http://23.253.211.234:8776/v2/",
          "rel": "self"
        }
      ],
      "status": "CURRENT",
      "updated": "2012-11-21T11:33:21Z"
    }
  ]
}
```

2.1.2. Show API v1 details

Method	URI	Description
GET	/v1	Shows Block Storage API v1 details.

Normal response codes: 200203

2.1.2.1. Request

This operation does not accept a request body.

2.1.2.2. Response

Example 2.2. Show API v1 details: JSON response

```
{
  "version": {
    "id": "v1.0",
    "links": [
      {
        "href": "http://23.253.211.234:8776/v1/",
        "rel": "self"
      },
      {
        "href": "http://docs.openstack.org/",
        "rel": "describedby",
        "type": "text/html"
      }
    ]
  },
  "media-types": [
    {
      "base": "application/xml",
      "type": "application/vnd.openstack.volume+xml;version=1"
    },
    {
      "base": "application/json",
      "type": "application/vnd.openstack.volume+json;version=1"
    }
  ],
  "status": "DEPRECATED",
  "updated": "2014-06-28T12:20:21Z"
}
```

2.2. Volumes

The `snapshot_id` and `source_vol_id` parameters specify the ID of the snapshot or volume from which the volume originates. If the volume was not created from a snapshot or source volume, these values are null.

Method	URI	Description
POST	/v1/{tenant_id}/volumes	Creates a volume.
GET	/v1/{tenant_id}/volumes	Lists all volumes.

Method	URI	Description
GET	/v1/{tenant_id}/volumes/detail	Lists all volumes, with details.
GET	/v1/{tenant_id}/volumes/{volume_id}	Shows details for a volume.
DELETE	/v1/{tenant_id}/volumes/{volume_id}	Deletes a volume.

2.2.1. Create volume

Method	URI	Description
POST	/v1/{tenant_id}/volumes	Creates a volume.

Normal response codes: 201

2.2.1.1. Request

This table shows the URI parameters for the create volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 2.3. Create volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume xmlns="http://docs.openstack.org/volume/api/v1"
  display_name="vol-001"
  display_description="Another volume."
  size="30"
  volume_type="289da7f8-6440-407c-9fb4-7db01ec49164"
  availability_zone="us-east1">
  <metadata>
    <meta key="contents">junk</meta>
  </metadata>
</volume>
```

Example 2.4. Create volume: JSON request

```
{
  "volume": {
    "display_name": "vol-001",
    "display_description": "Another volume.",
    "size": 30,
    "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
    "metadata": {
      "contents": "junk"
    },
    "availability_zone": "us-east1"
  }
}
```

This operation does not accept a request body.

2.2.1.2. Response

Example 2.5. Create volume: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume xmlns="http://docs.openstack.org/volume/api/v1"
  id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
  display_name="vol-001"
  display_description="Another volume."
  status="active">
```

```
size="30"
volume_type="289da7f8-6440-407c-9fb4-7db01ec49164"
availability_zone="us-east1"
bootable="false"
created_at="2012-02-14T20:53:07Z">
<metadata>
  <meta key="contents">junk</meta>
</metadata>
</volume>
```

Example 2.6. Create volume: JSON response

```
{
  "volume": {
    "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "display_name": "vol-001",
    "display_description": "Another volume.",
    "status": "active",
    "size": 30,
    "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
    "metadata": {
      "contents": "junk"
    },
    "availability_zone": "us-east1",
    "bootable": "false",
    "snapshot_id": null,
    "attachments": [
      {
        "attachment_id": "03987cd1-0ad5-40d1-9b2a-7cc48295d4fa",
        "id": "47e9ecc5-4045-4ee3-9a4b-d859d546a0cf",
        "volume_id": "6c80f8ac-e3e2-480c-8e6e-f1db92fe4bfe",
        "server_id": "d1c4788b-9435-42e2-9b81-29f3be1cd01f",
        "host_name": "mitaka",
        "device": "/"
      }
    ],
    "created_at": "2012-02-14T20:53:07Z"
  }
}
```

This operation does not return a response body.

2.2.2. List volumes

Method	URI	Description
GET	/v1/{tenant_id}/volumes	Lists all volumes.

Normal response codes: 200

2.2.2.1. Request

This table shows the URI parameters for the list volumes request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.2.2.2. Response

Example 2.7. List volumes: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volumes xmlns="http://docs.openstack.org/volume/api/v1">
  <volume xmlns="http://docs.openstack.org/volume/api/v1"
    id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
    display_name="vol-001"
    display_description="Another volume."
    status="active"
    size="30"
    volume_type="289da7f8-6440-407c-9fb4-7db01ec49164"
    availability_zone="us-east1"
    created_at="2012-02-14T20:53:07Z">
    <metadata>
      <meta key="contents">junk</meta>
    </metadata>
  </volume>
  <volume xmlns="http://docs.openstack.org/volume/api/v1"
    id="76b8950a-8594-4e5b-8dce-0dfa9c696358"
    display_name="vol-002"
    display_description="Yet another volume."
    status="active"
    size="25"
    volume_type="96c3bda7-c82a-4f50-be73-ca7621794835"
    availability_zone="us-east2"
    created_at="2012-03-15T19:10:03Z" />
</volumes>
```

Example 2.8. List volumes: JSON response

```
{
  "volumes": [
    {
      "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
      "display_name": "vol-001",
      "display_description": "Another volume.",
      "status": "active",
```



```

    "size": 30,
    "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
    "metadata": {
      "contents": "junk"
    },
    "availability_zone": "us-east1",
    "snapshot_id": null,
    "attachments": [
      {
        "attachment_id": "03987cd1-0ad5-40d1-9b2a-7cc48295d4fa",
        "id": "47e9ecc5-4045-4ee3-9a4b-d859d546a0cf",
        "volume_id": "6c80f8ac-e3e2-480c-8e6e-f1db92fe4bfe",
        "server_id": "d1c4788b-9435-42e2-9b81-29f3be1cd01f",
        "host_name": "mitaka",
        "device": "/"
      }
    ],
    "created_at": "2012-02-14T20:53:07Z"
  },
  {
    "id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
    "display_name": "vol-002",
    "display_description": "Yet another volume.",
    "status": "active",
    "size": 25,
    "volume_type": "96c3bda7-c82a-4f50-be73-ca7621794835",
    "metadata": {},
    "availability_zone": "us-east2",
    "snapshot_id": null,
    "attachments": [],
    "created_at": "2012-03-15T19:10:03Z"
  }
]
}

```

This operation does not return a response body.

2.2.3. List volumes, with details

Method	URI	Description
GET	/v1/{tenant_id}/volumes/detail	Lists all volumes, with details.

Normal response codes: 200

2.2.3.1. Request

This table shows the URI parameters for the list volumes, with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.2.3.2. Response

Example 2.9. List volumes, with details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volumes xmlns="http://docs.openstack.org/volume/api/v1">
  <volume xmlns="http://docs.openstack.org/volume/api/v1"
    id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
    display_name="vol-001"
    display_description="Another volume."
    status="active"
    size="30"
    volume_type="289da7f8-6440-407c-9fb4-7db01ec49164"
    availability_zone="us-east1"
    created_at="2012-02-14T20:53:07Z">
    <metadata>
      <meta key="contents">junk</meta>
    </metadata>
  </volume>
  <volume xmlns="http://docs.openstack.org/volume/api/v1"
    id="76b8950a-8594-4e5b-8dce-0dfa9c696358"
    display_name="vol-002"
    display_description="Yet another volume."
    status="active"
    size="25"
    volume_type="96c3bda7-c82a-4f50-be73-ca7621794835"
    availability_zone="us-east2"
    created_at="2012-03-15T19:10:03Z" />
</volumes>
```

Example 2.10. List volumes, with details: JSON response

```
{
  "volumes": [
    {
      "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
      "display_name": "vol-001",
      "display_description": "Another volume.",
      "status": "active",
```

```

    "size": 30,
    "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
    "metadata": {
      "contents": "junk"
    },
    "availability_zone": "us-east1",
    "snapshot_id": null,
    "attachments": [
      {
        "attachment_id": "03987cd1-0ad5-40d1-9b2a-7cc48295d4fa",
        "id": "47e9ecc5-4045-4ee3-9a4b-d859d546a0cf",
        "volume_id": "6c80f8ac-e3e2-480c-8e6e-f1db92fe4bfe",
        "server_id": "d1c4788b-9435-42e2-9b81-29f3be1cd01f",
        "host_name": "mitaka",
        "device": "/"
      }
    ],
    "created_at": "2012-02-14T20:53:07Z"
  },
  {
    "id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
    "display_name": "vol-002",
    "display_description": "Yet another volume.",
    "status": "active",
    "size": 25,
    "volume_type": "96c3bda7-c82a-4f50-be73-ca7621794835",
    "metadata": {},
    "availability_zone": "us-east2",
    "snapshot_id": null,
    "attachments": [],
    "created_at": "2012-03-15T19:10:03Z"
  }
]
}

```

This operation does not return a response body.

2.2.4. Show volume details

Method	URI	Description
GET	/v1/{tenant_id}/volumes/{volume_id}	Shows details for a volume.

Normal response codes: 200

2.2.4.1. Request

This table shows the URI parameters for the show volume details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

This operation does not accept a request body.

2.2.4.2. Response

Example 2.11. Show volume details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume xmlns="http://docs.openstack.org/volume/api/v1"
  id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
  display_name="vol-001"
  display_description="Another volume."
  status="active"
  size="30"
  volume_type="289da7f8-6440-407c-9fb4-7db01ec49164"
  availability_zone="us-east1"
  bootable="false"
  created_at="2012-02-14T20:53:07Z">
  <metadata>
    <meta key="contents">junk</meta>
  </metadata>
</volume>
```

Example 2.12. Show volume details: JSON response

```
{
  "volume": {
    "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "display_name": "vol-001",
    "display_description": "Another volume.",
    "status": "active",
    "size": 30,
    "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
    "metadata": {
      "contents": "junk"
    },
    "availability_zone": "us-east1",
    "bootable": "false",
    "snapshot_id": null,
    "attachments": [
```

```
    {
      "attachment_id": "03987cd1-0ad5-40d1-9b2a-7cc48295d4fa",
      "id": "47e9ecc5-4045-4ee3-9a4b-d859d546a0cf",
      "volume_id": "6c80f8ac-e3e2-480c-8e6e-f1db92fe4bfe",
      "server_id": "d1c4788b-9435-42e2-9b81-29f3be1cd01f",
      "host_name": "mitaka",
      "device": "/"
    }
  ],
  "created_at": "2012-02-14T20:53:07Z"
}
```

This operation does not return a response body.

2.2.5. Delete volume

Method	URI	Description
DELETE	/v1/{tenant_id}/volumes/{volume_id}	Deletes a volume.

Normal response codes: 202

2.2.5.1. Request

This table shows the URI parameters for the delete volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	UUID	The UUID of the volume.

This operation does not accept a request body.

2.3. Volume types

Lists, creates, updates, shows information for, and deletes volume types.

Method	URI	Description
GET	/v1/{tenant_id}/types	Lists volume types.
POST	/v1/{tenant_id}/types	Creates a volume type.
PUT	/v1/{tenant_id}/types/{volume_type_id}	Updates a volume type.
PUT	/v1/{tenant_id}/types/{volume_type_id}	Updates the extra specifications for a volume type.
GET	/v1/{tenant_id}/types/{volume_type_id}	Shows details for a volume type.
DELETE	/v1/{tenant_id}/types/{volume_type_id}	Deletes a volume type.

2.3.1. List volume types

Method	URI	Description
GET	/v1/{tenant_id}/types	Lists volume types.

Normal response codes: 200

2.3.1.1. Request

This table shows the URI parameters for the list volume types request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.3.1.2. Response

Example 2.13. List volume types: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_types xmlns="http://docs.openstack.org/volume/api/v1">
  <volume_type id="289da7f8-6440-407c-9fb4-7db01ec49164"
    name="vol-type-001">
    <extra_specs>
      <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
  </volume_type>
  <volume_type id="96c3bda7-c82a-4f50-be73-ca7621794835"
    name="vol-type-002" />
</volume_types>
```

Example 2.14. List volume types: JSON response

```
{
  "volume_types": [
    {
      "id": "289da7f8-6440-407c-9fb4-7db01ec49164",
      "name": "vol-type-001",
      "extra_specs": {
        "capabilities": "gpu"
      }
    },
    {
      "id": "96c3bda7-c82a-4f50-be73-ca7621794835",
      "name": "vol-type-002",
      "extra_specs": {}
    }
  ]
}
```

This operation does not return a response body.

2.3.2. Create volume type

Method	URI	Description
POST	/v1/{tenant_id}/types	Creates a volume type.

Normal response codes: 200

2.3.2.1. Request

This table shows the URI parameters for the create volume type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 2.15. Create volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
  name="vol-type-001">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 2.16. Create volume type: JSON request

```
{
  "volume_type": {
    "name": "vol-type-001",
    "extra_specs": {
      "capabilities": "gpu"
    }
  }
}
```

This operation does not accept a request body.

2.3.2.2. Response

Example 2.17. Create volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
  id="289da7f8-6440-407c-9fb4-7db01ec49164"
  name="vol-type-001">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 2.18. Create volume type: JSON response

```
{
  "volume_type": {
```



```
    "id": "289da7f8-6440-407c-9fb4-7db01ec49164",  
    "name": "vol-type-001",  
    "extra_specs": {  
        "capabilities": "gpu"  
    }  
}
```

This operation does not return a response body.

2.3.3. Update volume type

Method	URI	Description
PUT	/v1/{tenant_id}/types/{volume_type_id}	Updates a volume type.

Normal response codes: 200

2.3.3.1. Request

This table shows the URI parameters for the update volume type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_type_id}	UUID	The UUID for an existing volume type.

Example 2.19. Update volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
  name="vol-type-001">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 2.20. Update volume type: JSON request

```
{
  "volume_type": {
    "name": "vol-type-001",
    "extra_specs": {
      "capabilities": "gpu"
    }
  }
}
```

This operation does not accept a request body.

2.3.3.2. Response

Example 2.21. Update volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
  id="289da7f8-6440-407c-9fb4-7db01ec49164"
  name="vol-type-001">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 2.22. Update volume type: JSON response

```
{
```

```
"volume_type": {  
  "id": "289da7f8-6440-407c-9fb4-7db01ec49164",  
  "name": "vol-type-001",  
  "extra_specs": {  
    "capabilities": "gpu"  
  }  
}
```

This operation does not return a response body.

2.3.4. Update extra specs for a volume type

Method	URI	Description
PUT	/v1/{tenant_id}/types/{volume_type_id}	Updates the extra specifications for a volume type.

Normal response codes: 200

2.3.4.1. Request

This table shows the URI parameters for the update extra specs for a volume type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_type_id}	UUID	The UUID for an existing volume type.

Example 2.23. Update extra specs for a volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
  name="vol-type-001">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 2.24. Update extra specs for a volume type: JSON request

```
{
  "volume_type": {
    "name": "vol-type-001",
    "extra_specs": {
      "capabilities": "gpu"
    }
  }
}
```

This operation does not accept a request body.

2.3.4.2. Response

Example 2.25. Update extra specs for a volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
  id="289da7f8-6440-407c-9fb4-7db01ec49164"
  name="vol-type-001">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 2.26. Update extra specs for a volume type: JSON response

```
{
```

```
"volume_type": {  
  "id": "289da7f8-6440-407c-9fb4-7db01ec49164",  
  "name": "vol-type-001",  
  "extra_specs": {  
    "capabilities": "gpu"  
  }  
}
```

This operation does not return a response body.

2.3.5. Show volume type details

Method	URI	Description
GET	/v1/{tenant_id}/types/{volume_type_id}	Shows details for a volume type.

Normal response codes: 200

2.3.5.1. Request

This table shows the URI parameters for the show volume type details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_type_id}	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

2.3.5.2. Response

Example 2.27. Show volume type details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
  id="289da7f8-6440-407c-9fb4-7db01ec49164"
  name="vol-type-001">
  <extra_specs>
    <extra_spec key="capabilities">gpu</extra_spec>
  </extra_specs>
</volume_type>
```

Example 2.28. Show volume type details: JSON response

```
{
  "volume_type": {
    "id": "289da7f8-6440-407c-9fb4-7db01ec49164",
    "name": "vol-type-001",
    "extra_specs": {
      "capabilities": "gpu"
    }
  }
}
```

This operation does not return a response body.

2.3.6. Delete volume type

Method	URI	Description
DELETE	/v1/{tenant_id}/types/{volume_type_id}	Deletes a volume type.

Normal response codes: 202

2.3.6.1. Request

This table shows the URI parameters for the delete volume type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_type_id}	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

2.4. Snapshots

Creates, lists, shows information for, and deletes snapshots. Shows and updates snapshot metadata.

Method	URI	Description
POST	/v1/{tenant_id}/snapshots	Creates a snapshot.
GET	/v1/{tenant_id}/snapshots	Lists all snapshots.
GET	/v1/{tenant_id}/snapshots/detail	Lists all snapshots, with details.
GET	/v1/{tenant_id}/snapshots/{snapshot_id}	Shows details for a snapshot.
DELETE	/v1/{tenant_id}/snapshots/{snapshot_id}	Deletes a snapshot.
GET	/v1/{tenant_id}/snapshots/{snapshot_id}/metadata	Shows metadata for a snapshot.
PUT	/v1/{tenant_id}/snapshots/{snapshot_id}/metadata	Updates metadata for a snapshot.

2.4.1. Create snapshot

Method	URI	Description
POST	/v1/{tenant_id}/snapshots	Creates a snapshot.

Normal response codes: 201

2.4.1.1. Request

This table shows the URI parameters for the create snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 2.29. Create snapshot: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot xmlns="http://docs.openstack.org/volume/api/v1"
  name="snap-001" display_name="snap-001"
  display_description="Daily backup"
  volume_id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
  force="true"/>
```

Example 2.30. Create snapshot: JSON request

```
{
  "snapshot": {
    "display_name": "snap-001",
    "display_description": "Daily backup",
    "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "force": true
  }
}
```

This operation does not accept a request body.

2.4.1.2. Response

Example 2.31. Create snapshot: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot xmlns="http://docs.openstack.org/volume/api/v1"
  id="3fbbcccf-d058-4502-8844-6feeffdf4cb5"
  display_name="snap-001"
  display_description="Daily backup"
  volume_id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
  status="available"
  size="30"
  created_at="2012-02-29T03:50:07Z" />
```

Example 2.32. Create snapshot: JSON response

```
{
  "snapshot": {
```



```
    "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",  
    "display_name": "snap-001",  
    "display_description": "Daily backup",  
    "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",  
    "status": "available",  
    "size": 30,  
    "created_at": "2012-02-29T03:50:07Z"  
  }  
}
```

This operation does not return a response body.

2.4.2. List snapshots

Method	URI	Description
GET	/v1/{tenant_id}/snapshots	Lists all snapshots.

Normal response codes: 200

2.4.2.1. Request

This table shows the URI parameters for the list snapshots request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.4.2.2. Response

Example 2.33. List snapshots: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshots xmlns="http://docs.openstack.org/volume/api/v1">
  <snapshot id="3fbbcccf-d058-4502-8844-6feeffdf4cb5"
    display_name="snap-001"
    display_description="Daily backup"
    volume_id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
    status="available"
    size="30"
    created_at="2012-02-29T03:50:07Z">
    <metadata>
      <meta key="contents">junk</meta>
    </metadata>
  </snapshot>
  <snapshot id="e479997c-650b-40a4-9dfe-77655818b0d2"
    display_name="snap-002"
    display_description="Weekly backup"
    volume_id="76b8950a-8594-4e5b-8dce-0dfa9c696358"
    status="available"
    size="25"
    created_at="2012-03-19T01:52:47Z" />
</snapshots>
```

Example 2.34. List snapshots: JSON response

```
{
  "snapshots": [
    {
      "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
      "display_name": "snap-001",
      "display_description": "Daily backup",
      "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
      "status": "available",
      "size": 30,
      "created_at": "2012-02-29T03:50:07Z",
      "metadata": {
```

```
        "contents": "junk"
      }
    },
    {
      "id": "e479997c-650b-40a4-9dfe-77655818b0d2",
      "display_name": "snap-002",
      "display_description": "Weekly backup",
      "volume_id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
      "status": "available",
      "size": 25,
      "created_at": "2012-03-19T01:52:47Z",
      "metadata": {}
    }
  ]
}
```

This operation does not return a response body.

2.4.3. List snapshots with details

Method	URI	Description
GET	/v1/{tenant_id}/snapshots/detail	Lists all snapshots, with details.

Normal response codes: 200

2.4.3.1. Request

This table shows the URI parameters for the list snapshots with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.4.3.2. Response

Example 2.35. List snapshots with details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshots xmlns="http://docs.openstack.org/volume/api/v1">
  <snapshot id="3fbbcccf-d058-4502-8844-6feeffdf4cb5"
    display_name="snap-001"
    display_description="Daily backup"
    volume_id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
    status="available"
    size="30"
    created_at="2012-02-29T03:50:07Z">
    <metadata>
      <meta key="contents">junk</meta>
    </metadata>
  </snapshot>
  <snapshot id="e479997c-650b-40a4-9dfe-77655818b0d2"
    display_name="snap-002"
    display_description="Weekly backup"
    volume_id="76b8950a-8594-4e5b-8dce-0dfa9c696358"
    status="available"
    size="25"
    created_at="2012-03-19T01:52:47Z" />
</snapshots>
```

Example 2.36. List snapshots with details: JSON response

```
{
  "snapshots": [
    {
      "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
      "display_name": "snap-001",
      "display_description": "Daily backup",
      "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
      "status": "available",
      "size": 30,
      "created_at": "2012-02-29T03:50:07Z",
      "metadata": {
```

```
        "contents": "junk"
      }
    },
    {
      "id": "e479997c-650b-40a4-9dfe-77655818b0d2",
      "display_name": "snap-002",
      "display_description": "Weekly backup",
      "volume_id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
      "status": "available",
      "size": 25,
      "created_at": "2012-03-19T01:52:47Z",
      "metadata": {}
    }
  ]
}
```

This operation does not return a response body.

2.4.4. Show snapshot details

Method	URI	Description
GET	/v1/{tenant_id}/snapshots/{snapshot_id}	Shows details for a snapshot.

Normal response codes: 200

2.4.4.1. Request

This table shows the URI parameters for the show snapshot details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

2.4.4.2. Response

Example 2.37. Show snapshot details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot xmlns="http://docs.openstack.org/volume/api/v1"
  id="3fbbcccf-d058-4502-8844-6feeffdf4cb5"
  display_name="snap-001"
  display_description="Daily backup"
  volume_id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
  status="available"
  size="30"
  created_at="2012-02-29T03:50:07Z" />
```

Example 2.38. Show snapshot details: JSON response

```
{
  "snapshot": {
    "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
    "display_name": "snap-001",
    "display_description": "Daily backup",
    "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "status": "available",
    "size": 30,
    "created_at": "2012-02-29T03:50:07Z"
  }
}
```

This operation does not return a response body.

2.4.5. Delete snapshot

Method	URI	Description
DELETE	/v1/{tenant_id}/snapshots/{snapshot_id}	Deletes a snapshot.

Normal response codes: 202

2.4.5.1. Request

This table shows the URI parameters for the delete snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

2.4.6. Show snapshot metadata

Method	URI	Description
GET	/v1/{tenant_id}/snapshots/{snapshot_id}/metadata	Shows metadata for a snapshot.

Normal response codes: 200

2.4.6.1. Request

This table shows the URI parameters for the show snapshot metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

2.4.6.2. Response

Example 2.39. Show snapshot metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshot
  xmlns:os-extended-snapshot-attributes="http://docs.openstack.org/volume/
ext/extended_snapshot_attributes/api/v1"
  status="available" description="None"
  created_at="2014-05-06 17:59:52"
  volume_id="ebd80b99-bc3d-4154-9d28-5583baa80580" size="10"
  id="dfcd17fe-3b64-44ba-b95f-1c9c7109ef95" name="my-snapshot"
  os-extended-snapshot-attributes:project_id=
"7e0105e19cd2466193729ef78b604f79"
  os-extended-snapshot-attributes:progress="0%">
  <metadata>
    <meta key="key">v1</meta>
  </metadata>
</snapshot>
```

Example 2.40. Show snapshot metadata: JSON response

```
{
  "snapshot": {
    "status": "available",
    "os-extended-snapshot-attributes:progress": "0%",
    "description": null,
    "created_at": "2014-05-06T17:59:52.000000",
    "metadata": {
      "key": "v1"
    },
    "volume_id": "ebd80b99-bc3d-4154-9d28-5583baa80580",
    "os-extended-snapshot-attributes:project_id":
"7e0105e19cd2466193729ef78b604f79",
    "size": 10,
    "id": "dfcd17fe-3b64-44ba-b95f-1c9c7109ef95",
    "name": "my-snapshot"
  }
}
```



```
}  
}
```

This operation does not return a response body.

2.4.7. Update snapshot metadata

Method	URI	Description
PUT	/v1/{tenant_id}/snapshots/{snapshot_id}/metadata	Updates metadata for a snapshot.

Normal response codes: 200

2.4.7.1. Request

This table shows the URI parameters for the update snapshot metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

Example 2.41. Update snapshot metadata: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata>
  <meta key="key">v1</meta>
</metadata>
```

Example 2.42. Update snapshot metadata: JSON request

```
{
  "metadata": {
    "key": "v1"
  }
}
```

This operation does not accept a request body.

2.4.7.2. Response

Example 2.43. Update snapshot metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
  <meta key="key">v1</meta>
</metadata>
```

Example 2.44. Update snapshot metadata: JSON response

```
{
  "metadata": {
    "key": "v1"
  }
}
```

This operation does not return a response body.

2.5. Quota sets extension (os-quota-sets)

Administrators only, depending on policy settings.

Shows, updates, and deletes quotas for a tenant.

Method	URI	Description
GET	/v1/{tenant_id}/os-quota-sets/{tenant_id}{?usage}	Shows quotas for a tenant.
PUT	/v1/{tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.
DELETE	/v1/{tenant_id}/os-quota-sets/{tenant_id}	Deletes quotas for a tenant so the quotas revert to default values.
GET	/v1/{tenant_id}/os-quota-sets/default	Shows default quotas for a tenant.
GET	/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a tenant and user.
POST	/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a tenant and user.
DELETE	/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Deletes quotas for a user so that the quotas revert to default values.
GET	/v1/{tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}	Shows details for quotas for a tenant and user.

2.5.1. Show quotas

Method	URI	Description
GET	/v1/{tenant_id}/os-quota-sets/{tenant_id}{?usage}	Shows quotas for a tenant.

Normal response codes: 200

2.5.1.1. Request

This table shows the URI parameters for the show quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.

This table shows the query parameters for the show quotas request:

Name	Type	Description
usage	Boolean (Optional)	Set to usage=true to show quota usage. Default is false.

This operation does not accept a request body.

2.5.1.2. Response

Example 2.45. Show quotas response: JSON

```
{
  "quota_set": {
    "cores": 20,
    "fixed_ips": -1,
    "floating_ips": 10,
    "id": "fake_tenant",
    "injected_file_content_bytes": 10240,
    "injected_file_path_bytes": 255,
    "injected_files": 5,
    "instances": 10,
    "key_pairs": 100,
    "metadata_items": 128,
    "ram": 51200,
    "security_group_rules": 20,
    "security_groups": 10
  }
}
```

This table shows the body parameters for the show quotas response:

Name	Type	Description
quota_set	Dict (Required)	A quota_set object.
cores	Int	The number of instance cores that are allowed for each tenant.

Name	Type	Description
	<i>(Required)</i>	
fixed_ips	Int <i>(Required)</i>	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int <i>(Required)</i>	The number of floating IP addresses that are allowed for each tenant.
id	Int <i>(Required)</i>	The ID for the quota set.
injected_file_content_bytes	Int <i>(Required)</i>	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int <i>(Required)</i>	The number of bytes that are allowed for each injected file path.
injected_files	Int <i>(Required)</i>	The number of injected files that are allowed for each tenant.
instances	Int <i>(Required)</i>	The number of instances that are allowed for each tenant.
key_pairs	Int <i>(Required)</i>	The number of key pairs that are allowed for each user.
metadata_items	Int <i>(Required)</i>	The number of metadata items that are allowed for each instance.
ram	Int <i>(Required)</i>	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int <i>(Optional)</i>	The number of rules that are allowed for each security group.
security_groups	Int <i>(Required)</i>	The number of security groups that are allowed for each tenant.
in_use	String <i>(Optional)</i>	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int <i>(Optional)</i>	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 2.46. Show quotas response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <cores>20</cores>
  <fixed_ips>-1</fixed_ips>
  <floating_ips>10</floating_ips>
  <injected_file_content_bytes>10240</injected_file_content_bytes>
  <injected_file_path_bytes>255</injected_file_path_bytes>
  <injected_files>5</injected_files>
  <instances>10</instances>
  <key_pairs>100</key_pairs>
  <metadata_items>128</metadata_items>
  <ram>51200</ram>
  <security_group_rules>20</security_group_rules>
```

```
<security_groups>10</security_groups>  
</quota_set>
```

This operation does not return a response body.

2.5.2. Update quotas

Method	URI	Description
PUT	/v1/{tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.

Normal response codes: 200

2.5.2.1. Request

This table shows the URI parameters for the update quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.

Example 2.47. Update quotas response: JSON

```
{
  "quota_set": {
    "security_groups": 45
  }
}
```

This table shows the body parameters for the update quotas request:

Name	Type	Description
quota_set	Dict (Required)	A quota_set object.
cores	Int (Optional)	The number of instance cores that are allowed for each tenant.
fixed_ips	Int (Optional)	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int (Optional)	The number of floating IP addresses that are allowed for each tenant.
id	Int (Optional)	The ID for the quota set.
injected_file_content_bytes	Int (Optional)	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int (Optional)	The number of bytes that are allowed for each injected file path.
injected_files	Int (Optional)	The number of injected files that are allowed for each tenant.
instances	Int (Optional)	The number of instances that are allowed for each tenant.
key_pairs	Int	The number of key pairs that are allowed for each user.

Name	Type	Description
	<i>(Optional)</i>	
metadata_items	Int <i>(Optional)</i>	The number of metadata items that are allowed for each instance.
ram	Int <i>(Optional)</i>	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int <i>(Optional)</i>	The number of rules that are allowed for each security group.
security_groups	Int <i>(Optional)</i>	The number of security groups that are allowed for each tenant.

Example 2.48. Show quotas response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <security_groups>45</security_groups>
</quota_set>
```

This operation does not accept a request body.

2.5.2.2. Response

Example 2.49. Update quota response: JSON

```
{
  "quota_set": {
    "cores": 20,
    "fixed_ips": -1,
    "floating_ips": 10,
    "injected_file_content_bytes": 10240,
    "injected_file_path_bytes": 255,
    "injected_files": 5,
    "instances": 10,
    "key_pairs": 100,
    "metadata_items": 128,
    "ram": 51200,
    "security_group_rules": 20,
    "security_groups": 45
  }
}
```

This table shows the body parameters for the update quotas response:

Name	Type	Description
quota_set	Dict <i>(Required)</i>	A quota_set object.
cores	Int <i>(Required)</i>	The number of instance cores that are allowed for each tenant.
fixed_ips	Int <i>(Required)</i>	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int	The number of floating IP addresses that are allowed for each tenant.

Name	Type	Description
	(Required)	
id	Int (Required)	The ID for the quota set.
injected_file_content_bytes	Int (Required)	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes that are allowed for each injected file path.
injected_files	Int (Required)	The number of injected files that are allowed for each tenant.
instances	Int (Required)	The number of instances that are allowed for each tenant.
key_pairs	Int (Required)	The number of key pairs that are allowed for each user.
metadata_items	Int (Required)	The number of metadata items that are allowed for each instance.
ram	Int (Required)	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int (Optional)	The number of rules that are allowed for each security group.
security_groups	Int (Required)	The number of security groups that are allowed for each tenant.
in_use	String (Optional)	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int (Optional)	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 2.50. Update quota response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set>
  <cores>20</cores>
  <fixed_ips>-1</fixed_ips>
  <floating_ips>10</floating_ips>
  <injected_file_content_bytes>10240</injected_file_content_bytes>
  <injected_file_path_bytes>255</injected_file_path_bytes>
  <injected_files>5</injected_files>
  <instances>10</instances>
  <key_pairs>100</key_pairs>
  <metadata_items>128</metadata_items>
  <ram>51200</ram>
  <security_group_rules>20</security_group_rules>
  <security_groups>45</security_groups>
</quota_set>
```

This operation does not return a response body.

2.5.3. Delete quotas

Method	URI	Description
DELETE	/v1/{tenant_id}/os-quota-sets/{tenant_id}	Deletes quotas for a tenant so the quotas revert to default values.

Normal response codes: 200

2.5.3.1. Request

This table shows the URI parameters for the delete quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.

This operation does not accept a request body.

2.5.4. Show default quotas

Method	URI	Description
GET	/v1/{tenant_id}/os-quota-sets/default	Shows default quotas for a tenant.

Normal response codes: 200

2.5.4.1. Request

This table shows the URI parameters for the show default quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.5.4.2. Response

Example 2.51. Show default quotas: JSON response

```
{
  "quota_set": {
    "cores": 20,
    "fixed_ips": -1,
    "floating_ips": 10,
    "id": "fake_tenant",
    "injected_file_content_bytes": 10240,
    "injected_file_path_bytes": 255,
    "injected_files": 5,
    "instances": 10,
    "key_pairs": 100,
    "metadata_items": 128,
    "ram": 51200,
    "security_group_rules": 20,
    "security_groups": 10
  }
}
```

This table shows the body parameters for the show default quotas response:

Name	Type	Description
quota_set	Dict (Required)	A quota_set object.
cores	Int (Required)	The number of instance cores that are allowed for each tenant.
fixed_ips	Int (Required)	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int (Required)	The number of floating IP addresses that are allowed for each tenant.
id	Int	The ID for the quota set.

Name	Type	Description
	(Required)	
injected_file_content_bytes	Int (Required)	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes that are allowed for each injected file path.
injected_files	Int (Required)	The number of injected files that are allowed for each tenant.
instances	Int (Required)	The number of instances that are allowed for each tenant.
key_pairs	Int (Required)	The number of key pairs that are allowed for each user.
metadata_items	Int (Required)	The number of metadata items that are allowed for each instance.
ram	Int (Required)	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int (Optional)	The number of rules that are allowed for each security group.
security_groups	Int (Required)	The number of security groups that are allowed for each tenant.
in_use	String (Optional)	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int (Optional)	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 2.52. Show default quotas: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <cores>20</cores>
  <fixed_ips>-1</fixed_ips>
  <floating_ips>10</floating_ips>
  <injected_file_content_bytes>10240</injected_file_content_bytes>
  <injected_file_path_bytes>255</injected_file_path_bytes>
  <injected_files>5</injected_files>
  <instances>10</instances>
  <key_pairs>100</key_pairs>
  <metadata_items>128</metadata_items>
  <ram>51200</ram>
  <security_group_rules>20</security_group_rules>
  <security_groups>10</security_groups>
</quota_set>
```

This operation does not return a response body.

2.5.5. Show quotas for user

Method	URI	Description
GET	/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a tenant and user.

Normal response codes: 200

2.5.5.1. Request

This table shows the URI parameters for the show quotas for user request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{user_id}	String	The user ID. Specify in the URI as user_id={user_id}.

This operation does not accept a request body.

2.5.5.2. Response

Example 2.53. Show quotas for user response: JSON

```
{
  "quota_set": {
    "cores": 20,
    "fixed_ips": -1,
    "floating_ips": 10,
    "id": "fake_tenant",
    "injected_file_content_bytes": 10240,
    "injected_file_path_bytes": 255,
    "injected_files": 5,
    "instances": 10,
    "key_pairs": 100,
    "metadata_items": 128,
    "ram": 51200,
    "security_group_rules": 20,
    "security_groups": 10
  }
}
```

This table shows the body parameters for the show quotas for user response:

Name	Type	Description
quota_set	Dict (Required)	A quota_set object.
cores	Int (Required)	The number of instance cores that are allowed for each tenant.
fixed_ips	Int (Required)	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int	The number of floating IP addresses that are allowed for each tenant.

Name	Type	Description
	(Required)	
id	Int (Required)	The ID for the quota set.
injected_file_content_bytes	Int (Required)	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes that are allowed for each injected file path.
injected_files	Int (Required)	The number of injected files that are allowed for each tenant.
instances	Int (Required)	The number of instances that are allowed for each tenant.
key_pairs	Int (Required)	The number of key pairs that are allowed for each user.
metadata_items	Int (Required)	The number of metadata items that are allowed for each instance.
ram	Int (Required)	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int (Optional)	The number of rules that are allowed for each security group.
security_groups	Int (Required)	The number of security groups that are allowed for each tenant.
in_use	String (Optional)	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int (Optional)	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 2.54. Show quotas for user response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <cores>20</cores>
  <fixed_ips>-1</fixed_ips>
  <floating_ips>10</floating_ips>
  <injected_file_content_bytes>10240</injected_file_content_bytes>
  <injected_file_path_bytes>255</injected_file_path_bytes>
  <injected_files>5</injected_files>
  <instances>10</instances>
  <key_pairs>100</key_pairs>
  <metadata_items>128</metadata_items>
  <ram>51200</ram>
  <security_group_rules>20</security_group_rules>
  <security_groups>10</security_groups>
</quota_set>
```

This operation does not return a response body.

2.5.6. Update quotas for user

Method	URI	Description
POST	/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a tenant and user.

Normal response codes: 200

2.5.6.1. Request

This table shows the URI parameters for the update quotas for user request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{user_id}	String	The user ID. Specify in the URI as user_id={user_id}.

Example 2.55. Update quotas for user request: JSON

```
{
  "quota_set": {
    "force": true,
    "instances": 9
  }
}
```

This table shows the body parameters for the update quotas for user request:

Name	Type	Description
quota_set	Dict (Required)	A quota_set object.
cores	Int (Optional)	The number of instance cores that are allowed for each tenant.
fixed_ips	Int (Optional)	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int (Optional)	The number of floating IP addresses that are allowed for each tenant.
id	Int (Optional)	The ID for the quota set.
injected_file_content_bytes	Int (Optional)	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int (Optional)	The number of bytes that are allowed for each injected file path.
injected_files	Int (Optional)	The number of injected files that are allowed for each tenant.
instances	Int (Optional)	The number of instances that are allowed for each tenant.
key_pairs	Int	The number of key pairs that are allowed for each user.

Name	Type	Description
	<i>(Optional)</i>	
metadata_items	Int <i>(Optional)</i>	The number of metadata items that are allowed for each instance.
ram	Int <i>(Optional)</i>	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int <i>(Optional)</i>	The number of rules that are allowed for each security group.
security_groups	Int <i>(Optional)</i>	The number of security groups that are allowed for each tenant.

Example 2.56. Update quotas for user request: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <force>true</force>
  <instances>9</instances>
</quota_set>
```

This operation does not accept a request body.

2.5.6.2. Response

Example 2.57. Update quotas for user response: JSON

```
{
  "quota_set": {
    "cores": 20,
    "floating_ips": 10,
    "fixed_ips": -1,
    "injected_file_content_bytes": 10240,
    "injected_file_path_bytes": 255,
    "injected_files": 5,
    "instances": 9,
    "key_pairs": 100,
    "metadata_items": 128,
    "ram": 51200,
    "security_group_rules": 20,
    "security_groups": 10
  }
}
```

This table shows the body parameters for the update quotas for user response:

Name	Type	Description
quota_set	Dict <i>(Required)</i>	A quota_set object.
cores	Int <i>(Required)</i>	The number of instance cores that are allowed for each tenant.
fixed_ips	Int <i>(Required)</i>	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.

Name	Type	Description
floating_ips	Int (Required)	The number of floating IP addresses that are allowed for each tenant.
id	Int (Required)	The ID for the quota set.
injected_file_content_bytes	Int (Required)	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes that are allowed for each injected file path.
injected_files	Int (Required)	The number of injected files that are allowed for each tenant.
instances	Int (Required)	The number of instances that are allowed for each tenant.
key_pairs	Int (Required)	The number of key pairs that are allowed for each user.
metadata_items	Int (Required)	The number of metadata items that are allowed for each instance.
ram	Int (Required)	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int (Optional)	The number of rules that are allowed for each security group.
security_groups	Int (Required)	The number of security groups that are allowed for each tenant.
in_use	String (Optional)	The in use data size. Visible only if you set the usage=true query parameter.
reserved	Int (Optional)	Reserved volume size. Visible only if you set the usage=true query parameter.

Example 2.58. Show quotas for user response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set>
  <cores>20</cores>
  <floating_ips>10</floating_ips>
  <fixed_ips>-1</fixed_ips>
  <injected_file_content_bytes>10240</injected_file_content_bytes>
  <injected_file_path_bytes>255</injected_file_path_bytes>
  <injected_files>5</injected_files>
  <instances>9</instances>
  <key_pairs>100</key_pairs>
  <metadata_items>128</metadata_items>
  <ram>51200</ram>
  <security_group_rules>20</security_group_rules>
  <security_groups>10</security_groups>
</quota_set>
```

This operation does not return a response body.

2.5.7. Delete quotas for user

Method	URI	Description
DELETE	/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Deletes quotas for a user so that the quotas revert to default values.

Normal response codes: 200

2.5.7.1. Request

This table shows the URI parameters for the delete quotas for user request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{user_id}	String	The user ID. Specify in the URI as user_id={user_id}.

This operation does not accept a request body.

2.5.8. Show quota details for user

Method	URI	Description
GET	/v1/{tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}	Shows details for quotas for a tenant and user.

Normal response codes: 200

2.5.8.1. Request

This table shows the URI parameters for the show quota details for user request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{user_id}	String	The user ID. Specify in the URI as user_id={user_id}.

This operation does not accept a request body.

2.5.8.2. Response

Example 2.59. Show quota details for user response: JSON

```
{
  "quota_set": {
    "cores": {
      "in_use": 0,
      "limit": 20,
      "reserved": 0
    },
    "fixed_ips": {
      "in_use": 0,
      "limit": -1,
      "reserved": 0
    },
    "floating_ips": {
      "in_use": 0,
      "limit": 10,
      "reserved": 0
    },
    "injected_files": {
      "in_use": 0,
      "limit": 5,
      "reserved": 0
    },
    "instances": {
      "in_use": 0,
      "limit": 10,
      "reserved": 0
    },
    "key_pairs": {
      "in_use": 0,
      "limit": 100,
      "reserved": 0
    }
  }
}
```

```
    "metadata_items": {
      "in_use": 0,
      "limit": 128,
      "reserved": 0
    },
    "ram": {
      "in_use": 0,
      "limit": 51200,
      "reserved": 0
    },
    "security_groups": {
      "in_use": 0,
      "limit": 10,
      "reserved": 0
    },
    "injected_file_content_bytes": {
      "in_use": 0,
      "limit": 10240,
      "reserved": 0
    },
    "injected_file_path_bytes": {
      "in_use": 0,
      "limit": 255,
      "reserved": 0
    },
    "security_group_rules": {
      "in_use": 0,
      "limit": 20,
      "reserved": 0
    }
  }
}
```

3. Clustering API v1 (CURRENT)

Clustering service for OpenStack.

For more information, see [Senlin](#).

Method	URI	Description
API versions		
GET	/	Lists information for all Clustering API versions.
Actions		
GET	/v1/actions{?limit,marker,sort,global_project,name,target,action}	Lists all actions.
GET	/v1/actions/{action_id}	Shows details for an action.
Build information		
GET	/v1/build-info	Shows build information for a Senlin deployment.
Clusters		
GET	/v1/clusters{?limit,marker,sort,global_project,name,status}	Lists clusters.
POST	/v1/clusters	Creates a cluster.
GET	/v1/clusters/{cluster_id}	Shows details for a cluster.
PATCH	/v1/clusters/{cluster_id}	Updates a cluster.
DELETE	/v1/clusters/{cluster_id}	Deletes a cluster.
POST	/v1/clusters/{cluster_id}/actions	Triggers an action on a cluster.
Cluster policies		
GET	/v1/clusters/{cluster_id}/policies{?sort,enabled}	Lists all policies for a cluster.
GET	/v1/clusters/{cluster_id}/policies/{policy_id}	Shows details for a policy for a cluster.
Events		
GET	/v1/events{?limit,marker,sort,global_project,obj_id,obj_type,obj_name,cluster_id,action}	Lists events.
GET	/v1/events/{event_id}	Shows details for an event.
Nodes		
GET	/v1/nodes{?limit,marker,sort,global_project,cluster_id,name,status}	Lists all nodes.
POST	/v1/nodes	Creates a node.
GET	/v1/nodes/{node_id}{?show_details}	Shows data for a node.
PATCH	/v1/nodes/{node_id}	Updates a node.
DELETE	/v1/nodes/{node_id}	Deletes a node.
POST	/v1/nodes/{node_id}/actions	Triggers an action on a node.
Policies		
GET	/v1/policies{?limit,marker,sort,global_project,name,type}	Lists all policies.
POST	/v1/policies	Creates a policy.
GET	/v1/policies/{policy_id}	Shows details for a policy.
PATCH	/v1/policies/{policy_id}	Updates a policy.
DELETE	/v1/policies/{policy_id}	Deletes a policy.

Method	URI	Description
Policy types		
GET	/v1/policy-types	Lists all supported policy types.
GET	/v1/policy-types/{policy_type}	Shows details for a policy type.
Profiles		
GET	/v1/profiles{?limit,marker,sort,global_project,name,type,metadata}	Lists all profiles.
POST	/v1/profiles	Creates a profile.
GET	/v1/profiles/{profile_id}	Shows details for a profile.
PATCH	/v1/profiles/{profile_id}	Updates a profile.
DELETE	/v1/profiles/{profile_id}	Deletes a profile.
Profile types		
GET	/v1/profile-types	Lists supported profile types.
GET	/v1/profile-types/{profile_type}	Shows details for a profile type.
Receivers		
GET	/v1/receivers{?limit,marker,sort,global_project,name,type,user,cluster_id,action}	Lists all receivers.
POST	/v1/receivers	Creates a receiver.
GET	/v1/receivers/{receiver_id}	Shows details for a receiver.
DELETE	/v1/receivers/{receiver_id}	Deletes a receiver.
Webhooks		
POST	/v1/webhooks/{webhook_id}/trigger{?V,params}	Triggers a webhook receiver.

3.1. API versions

Lists information for all Clustering API versions.

Method	URI	Description
GET	/	Lists information for all Clustering API versions.

3.1.1. List versions

Method	URI	Description
GET	/	Lists information for all Clustering API versions.

Normal response codes: 200

3.1.1.1. Request

This operation does not accept a request body.

3.1.1.2. Response

Example 3.1. List versions: JSON response

```
{
  "versions": [
    {
      "status": "CURRENT",
      "id": "v1.0",
      "links": [
        {
          "href": "http://192.168.12.34:8778/v1/",
          "rel": "self"
        }
      ]
    }
  ]
}
```

3.2. Actions

Lists all actions and shows details for an action.

Method	URI	Description
GET	/v1/actions{?limit,marker,sort,global_project,name,target,action}	Lists all actions.
GET	/v1/actions/{action_id}	Shows details for an action.

3.2.1. List actions

Method	URI	Description
GET	/v1/actions{?limit,marker,sort,global_project,name,target,action}	Lists all actions.

Normal response codes: 200

3.2.1.1. Request

This table shows the query parameters for the list actions request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
sort	String (Optional)	Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is <code>asc</code> (ascending) or <code>desc</code> (descending). Default direction is <code>asc</code> (ascending). Specify the list as <code><key>[:<direction>]</code> . For example, the following query parameters in the URI sort the objects in the response by name in ascending order and then by <code>status</code> in descending order: <code>GET /v2/images?sort=name:asc,status:desc</code> The following query parameters in the URI sort the objects in the response by name in descending order and then by <code>status</code> in ascending order. <code>GET /v2/images?sort=name:desc,status</code>
global_project	Boolean (Optional)	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to <code>true</code> , the call returns all objects from all projects. Default is <code>false</code> , which returns only objects in the current project.
name	String (Optional)	Filters the response by an action name. Use this filter multiple times to filter by multiple names.
target	String (Optional)	Filters the response by the targeted object ID that is associated with an action. An object can be a cluster, a node, and so on. Use this filter multiple times to filter by multiple targets.
action	String (Optional)	Filters the response by an action name. Use this filter multiple times to filter by multiple names.

This operation does not accept a request body.

3.2.1.2. Response

Example 3.2. List actions: JSON response

An actions list that contains all records that meet the criteria. Each record provides detailed data about an action.

```
{
  "actions": [
    {
      "action": "NODE_CREATE",
      "cause": "RPC Request",
      "created_at": "2015-12-04T04:54:41",
      "depended_by": [],
      "depends_on": [],
      "end_time": 1425550000.0,
      "id": "2366d440-c73e-4961-9254-6d1c3af7c167",
      "inputs": {},
      "interval": -1,
      "name": "node_create_0df0931b",
      "outputs": {},
      "owner": null,
      "start_time": 1425550000.0,
      "status": "SUCCEEDED",
      "status_reason": "Action completed successfully.",
      "target": "0df0931b-e251-4f2e-8719-4ebfda3627ba",
      "timeout": 3600,
      "updated_at": null
    },
    {
      "action": "NODE_DELETE",
      "cause": "RPC Request",
      "created_at": "2015-11-04T05:21:41",
      "depended_by": [],
      "depends_on": [],
      "end_time": 1425550000.0,
      "id": "edce3528-864f-41fb-8759-f4707925cc09",
      "inputs": {},
      "interval": -1,
      "name": "node_delete_f0de9b9c",
      "outputs": {},
      "owner": null,
      "start_time": 1425550000.0,
      "status": "SUCCEEDED",
      "status_reason": "Action completed successfully.",
      "target": "f0de9b9c-6d48-4a46-af21-2ca8607777fe",
      "timeout": 3600,
      "updated_at": null
    }
  ]
}
```

3.2.2. Show action details

Method	URI	Description
GET	/v1/actions/{action_id}	Shows details for an action.

Normal response codes: 200

3.2.2.1. Request

This table shows the URI parameters for the show action details request:

Name	Type	Description
{action_id}	UUID	The UUID of the action.

This operation does not accept a request body.

3.2.2.2. Response

Example 3.3. Show action details: JSON response

An action map. Contains fields such as `id`, `action`, and so on.

```
{
  "action": "CLUSTER_DELETE",
  "cause": "RPC Request",
  "context": {},
  "created_at": "2015-06-27T05:09:43",
  "depended_by": [],
  "depends_on": [],
  "end_time": 1423570000.0,
  "id": "ffbb9175-d510-4bc1-b676-c6aba2a4ca81",
  "inputs": {},
  "interval": -1,
  "name": "cluster_delete_fcc9b635",
  "outputs": {},
  "owner": null,
  "start_time": 1423570000.0,
  "status": "FAILED",
  "status_reason": "Cluster action FAILED",
  "target": "fcc9b635-52e3-490b-99f2-87b1640e4e89",
  "timeout": 3600,
  "updated_at": null
}
```

3.3. Build information

Shows build information for a [Senlin](#) deployment.

Method	URI	Description
GET	/v1/build-info	Shows build information for a Senlin deployment.

3.3.1. Show build information

Method	URI	Description
GET	/v1/build-info	Shows build information for a Senlin deployment.

Normal response codes: 200

3.3.1.1. Request

This operation does not accept a request body.

3.3.1.2. Response

Example 3.4. Show build information: JSON response

```
{
  "build_info": {
    "api": {
      "revision": "1.0"
    },
    "engine": {
      "revision": "2.0"
    }
  }
}
```

3.4. Clusters

Lists all clusters and creates, shows information for, updates, deletes, and triggers an action on a cluster.

Method	URI	Description
GET	/v1/clusters{?limit,marker,sort,global_project,name,status}	Lists clusters.
POST	/v1/clusters	Creates a cluster.
GET	/v1/clusters/{cluster_id}	Shows details for a cluster.
PATCH	/v1/clusters/{cluster_id}	Updates a cluster.
DELETE	/v1/clusters/{cluster_id}	Deletes a cluster.
POST	/v1/clusters/{cluster_id}/actions	Triggers an action on a cluster.

3.4.1. List clusters

Method	URI	Description
GET	/v1/clusters{?limit,marker,sort,global_project,name,status}	Lists clusters.

Normal response codes: 200

3.4.1.1. Request

This table shows the query parameters for the list clusters request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
sort	String (Optional)	Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is <code>asc</code> (ascending) or <code>desc</code> (descending). Default direction is <code>asc</code> (ascending). Specify the list as <code><key>[:<direction>]</code> . For example, the following query parameters in the URI sort the objects in the response by name in ascending order and then by status in descending order: <code>GET /v2/images?sort=name:asc,status:desc</code> The following query parameters in the URI sort the objects in the response by name in descending order and then by status in ascending order. <code>GET /v2/images?sort=name:desc,status</code>
global_project	Boolean (Optional)	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to <code>true</code> , the call returns all objects from all projects. Default is <code>false</code> , which returns only objects in the current project.
name	String (Optional)	Filters the response by a cluster name. Use this filter multiple times to filter by multiple names.
status	String (Optional)	Filters the response by a cluster status. Use this filter multiple times to filter by multiple statuses.

This operation does not accept a request body.

3.4.1.2. Response

Example 3.5. List clusters: JSON response

```
{
  "clusters": [
```

```
{
  "created_at": "2015-02-10T14:26:14",
  "data": {},
  "desired_capacity": 4,
  "domain": null,
  "id": "7d85f602-a948-4a30-afd4-e84f47471c15",
  "init_at": "2015-02-10T14:26:11",
  "max_size": -1,
  "metadata": {},
  "min_size": 0,
  "name": "cluster1",
  "nodes": [
    "b07c57c8-7ab2-47bf-bdf8-e894c0c601b9",
    "ecc23d3e-bb68-48f8-8260-c9cf6bcb6e61",
    "da1e9c87-e584-4626-a120-022da5062dac"
  ],
  "policies": [],
  "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
  "profile_name": "mystack",
  "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
  "status": "ACTIVE",
  "status_reason": "Cluster scale-in succeeded",
  "timeout": 3600,
  "updated_at": null,
  "user": "5e5bf8027826429c96af157f68dc9072"
}
]
```

3.4.2. Create cluster

Method	URI	Description
POST	/v1/clusters	Creates a cluster.

Normal response codes: 202

3.4.2.1. Request

Example 3.6. Create cluster: JSON request

```
{
  "cluster": {
    "desired_capacity": 0,
    "max_size": -1,
    "metadata": {},
    "min_size": 0,
    "name": "test_cluster",
    "profile_id": "mystack",
    "timeout": null
  }
}
```

3.4.2.2. Response

Example 3.7. Create cluster: JSON response

```
{
  "cluster": {
    "action": "bbf4558b-9fa3-482a-93c2-a4aa5cc85317",
    "created_at": null,
    "data": {},
    "desired_capacity": 4,
    "domain": null,
    "id": "45edadcb-c73b-4920-87e1-518b2f29f54b",
    "init_at": "2015-02-10T14:16:10",
    "max_size": -1,
    "metadata": {},
    "min_size": 0,
    "name": "test_cluster",
    "nodes": [],
    "policies": [],
    "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
    "profile_name": "mystack",
    "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
    "status": "INIT",
    "status_reason": "Initializing",
    "timeout": 3600,
    "updated_at": null,
    "user": "5e5bf8027826429c96af157f68dc9072"
  }
}
```

3.4.3. Show cluster details

Method	URI	Description
GET	/v1/clusters/{cluster_id}	Shows details for a cluster.

Normal response codes: 200

3.4.3.1. Request

This table shows the URI parameters for the show cluster details request:

Name	Type	Description
{cluster_id}	UUID	The UUID of the cluster.

This operation does not accept a request body.

3.4.3.2. Response

Example 3.8. Show cluster details: JSON response

```
{
  "cluster": {
    "created_at": "2015-02-11T15:13:20",
    "data": {},
    "desired_capacity": 0,
    "domain": null,
    "id": "45edadcb-c73b-4920-87e1-518b2f29f54b",
    "init_at": "2015-02-10T14:26:10",
    "max_size": -1,
    "metadata": {},
    "min_size": 0,
    "name": "test_cluster",
    "nodes": [],
    "policies": [],
    "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
    "profile_name": "mystack",
    "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
    "status": "ACTIVE",
    "status_reason": "Creation succeeded",
    "timeout": 3600,
    "updated_at": null,
    "user": "5e5bf8027826429c96af157f68dc9072"
  }
}
```

3.4.4. Update cluster

Method	URI	Description
PATCH	/v1/clusters/{cluster_id}	Updates a cluster.

Normal response codes: 202

3.4.4.1. Request

This table shows the URI parameters for the update cluster request:

Name	Type	Description
{cluster_id}	UUID	The UUID of the cluster.

Example 3.9. Update cluster: JSON request

```
{
  "cluster": {
    "metadata": null,
    "name": null,
    "profile_id": null,
    "timeout": "30"
  }
}
```

3.4.4.2. Response

Example 3.10. Update cluster: JSON response

```
{
  "cluster": {
    "created_at": "2015-02-11T15:13:20",
    "data": {},
    "desired_capacity": 0,
    "domain": null,
    "id": "45edadcb-c73b-4920-87e1-518b2f29f54b",
    "init_at": "2015-02-10T14:26:10",
    "max_size": -1,
    "metadata": {},
    "min_size": 0,
    "name": "test_cluster",
    "nodes": [],
    "policies": [],
    "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
    "profile_name": "mystack",
    "project": "6e18cc2bdb48a5b3cad2dc499f6804",
    "status": "INIT",
    "status_reason": "Initializing",
    "timeout": 3600,
    "updated_at": null,
    "user": "5e5bf8027826429c96af157f68dc9072"
  }
}
```


3.4.5. Delete cluster

Method	URI	Description
DELETE	/v1/clusters/{cluster_id}	Deletes a cluster.

Normal response codes: 202

3.4.5.1. Request

This table shows the URI parameters for the delete cluster request:

Name	Type	Description
{cluster_id}	UUID	The UUID of the cluster.

This operation does not accept a request body.

3.4.5.2. Response

The Location response header contains the action that the request triggered.

This operation does not return a response body.

3.4.6. Trigger cluster action

Method	URI	Description
POST	/v1/clusters/{cluster_id}/actions	Triggers an action on a cluster.

Normal response codes: 202

3.4.6.1. Request

This table shows the URI parameters for the trigger cluster action request:

Name	Type	Description
{cluster_id}	UUID	The UUID of the cluster.

3.4.6.2. Response

The Location response header contains the action that the request initiates.

This operation does not return a response body.

3.5. Cluster policies

Lists all cluster policies and shows information for a cluster policy.

Method	URI	Description
GET	/v1/clusters/{cluster_id}/policies{?sort,enabled}	Lists all policies for a cluster.
GET	/v1/clusters/{cluster_id}/policies/{policy_id}	Shows details for a policy for a cluster.

3.5.1. List policies

Method	URI	Description
GET	/v1/clusters/{cluster_id}/policies {?sort,enabled}	Lists all policies for a cluster.

Normal response codes: 200

3.5.1.1. Request

This table shows the URI parameters for the list policies request:

Name	Type	Description
{cluster_id}	UUID	The UUID of the cluster.

This table shows the query parameters for the list policies request:

Name	Type	Description
sort	String (Optional)	Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is asc (ascending) or desc (descending). Default direction is asc (ascending). Specify the list as <key>[:<direction>]. For example, the following query parameters in the URI sort the objects in the response by name in ascending order and then by status in descending order: <pre>GET /v2/images?sort=name:asc,status:desc</pre> The following query parameters in the URI sort the objects in the response by name in descending order and then by status in ascending order. <pre>GET /v2/images?sort=name:desc,status</pre>
enabled	Boolean (Optional)	Filters the response by a policy enabled status on the cluster.

This operation does not accept a request body.

3.5.1.2. Response

Example 3.11. List policies: JSON response

A `cluster_policies` list contains records of cluster-policy bindings. Each record contains cluster-related fields, the policy, and the binding itself.

```
{
  "cluster_policies": [
    {
      "cluster_id": "7d85f602-a948-4a30-afd4-e84f47471c15",
      "cluster_name": "cluster4",
      "enabled": true,
      "id": "06be3a1f-b238-4a96-a737-ceec5714087e",
      "policy_id": "714fe676-a08f-4196-b7af-61d52eeded15",
      "policy_name": "dp01",

```

```
        "policy_type": "senlin.policy.deletion-1.0"
    },
    {
        "cluster_id": "7d85f602-a948-4a30-afd4-e84f47471c15",
        "cluster_name": "cluster4",
        "enabled": true,
        "id": "abddc45e-ac31-4f90-93cc-db55a7d8dd6d",
        "policy_id": "e026e09f-a3e9-4dad-a1b9-d7ba316026a1",
        "policy_name": "sp1",
        "policy_type": "senlin.policy.scaling-1.0"
    }
]
}
```

3.5.2. Show cluster policy details

Method	URI	Description
GET	/v1/clusters/{cluster_id}/policies/{policy_id}	Shows details for a policy for a cluster.

Normal response codes: 200

3.5.2.1. Request

This table shows the URI parameters for the show cluster policy details request:

Name	Type	Description
{cluster_id}	UUID	The UUID of the cluster.
{policy_id}	UUID	The UUID of the policy.

This operation does not accept a request body.

3.5.2.2. Response

Example 3.12. Show cluster policy details: JSON response

A `cluster_policy` map with keys such as `cluster_id`, `policy_id`, `policy_type`, and so on. The key values are attributes for the policy that is attached to the cluster.

```
{
  "cluster_policy": {
    "cluster_id": "7d85f602-a948-4a30-afd4-e84f47471c15",
    "cluster_name": "cluster4",
    "enabled": true,
    "id": "06be3a1f-b238-4a96-a737-ceec5714087e",
    "policy_id": "714fe676-a08f-4196-b7af-61d52eeded15",
    "policy_name": "dp01",
    "policy_type": "senlin.policy.deletion-1.0"
  }
}
```

3.6. Events

Lists all events and shows information for an event.

Method	URI	Description
GET	/v1/events{?limit,marker,sort,global_project,obj_id,obj_type,obj_name,cluster_id,action}	Lists events.
GET	/v1/events/{event_id}	Shows details for an event.

3.6.1. List events

Method	URI	Description
GET	/v1/events{?limit,marker,sort,global_project,obj_id,obj_type,obj_name,cluster_id,action}	Lists events.

Normal response codes: 200

3.6.1.1. Request

This table shows the query parameters for the list events request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
sort	String (Optional)	Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is <code>asc</code> (ascending) or <code>desc</code> (descending). Default direction is <code>asc</code> (ascending). Specify the list as <code><key>[:<direction>]</code> . For example, the following query parameters in the URI sort the objects in the response by name in ascending order and then by status in descending order: <code>GET /v2/images?sort=name:asc,status:desc</code> The following query parameters in the URI sort the objects in the response by name in descending order and then by status in ascending order. <code>GET /v2/images?sort=name:desc,status</code>
global_project	Boolean (Optional)	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to <code>true</code> , the call returns all objects from all projects. Default is <code>false</code> , which returns only objects in the current project.
obj_id	String (Optional)	Filters the response by the object ID for an event. Use this filter multiple times to filter by multiple objects.
obj_type	String (Optional)	Filters the response by the type of object associated with an event. Use this filter multiple times to filter by multiple objects. A valid value is <code>CLUSTER</code> or <code>NODE</code> .
obj_name	String (Optional)	Filters the response by the name of object associated with an event. Use this filter multiple times to filter by multiple objects.
cluster_id	String (Optional)	Filters the response by the cluster ID associated with an event. Use this filter multiple times to filter by multiple clusters.
action	String (Optional)	Filters the response by the action name associated with an event. Use this filter multiple times to filter by multiple actions.

This operation does not accept a request body.

3.6.1.2. Response

Example 3.13. List events: JSON response

```
{
  "events": [
    {
      "action": "create",
      "cluster_id": null,
      "id": "2d255b9c-8f36-41a2-a137-c0175ccc29c3",
      "level": "20",
      "obj_id": "0df0931b-e251-4f2e-8719-4ebfda3627ba",
      "obj_name": "node009",
      "obj_type": "NODE",
      "project": "6e18cc2bdb48a5b3cad2dc499f6804",
      "status": "CREATING",
      "status_reason": "Initializing",
      "timestamp": "2015-03-05T08:53:15",
      "user": "a21ded6060534d99840658a777c2af5a"
    }
  ]
}
```

3.6.2. Show event details

Method	URI	Description
GET	/v1/events/{event_id}	Shows details for an event.

Normal response codes: 200

3.6.2.1. Request

This table shows the URI parameters for the show event details request:

Name	Type	Description
{event_id}	UUID	The UUID of the event.

This operation does not accept a request body.

3.6.2.2. Response

Example 3.14. Show event details: JSON response

An event map, which contains the details of an event such as `TIMESTAMP` or `LEVEL`.

```
{
  "event": {
    "action": "create",
    "cluster_id": null,
    "id": "2d255b9c-8f36-41a2-a137-c0175ccc29c3",
    "level": "20",
    "obj_id": "0df0931b-e251-4f2e-8719-4ebfda3627ba",
    "obj_name": "node009",
    "obj_type": "NODE",
    "project": "6e18cc2bdb48a5b3cad2dc499f6804",
    "status": "CREATING",
    "status_reason": "Initializing",
    "timestamp": "2015-03-05T08:53:15",
    "user": "a21ded6060534d99840658a777c2af5a"
  }
}
```

3.7. Nodes

Lists all nodes, and creates, shows information for, updates, deletes a node.

Method	URI	Description
GET	/v1/nodes{?limit,marker,sort,global_project,cluster_id,name,status}	Lists all nodes.
POST	/v1/nodes	Creates a node.
GET	/v1/nodes/{node_id}{?show_details}	Shows data for a node.
PATCH	/v1/nodes/{node_id}	Updates a node.
DELETE	/v1/nodes/{node_id}	Deletes a node.
POST	/v1/nodes/{node_id}/actions	Triggers an action on a node.

3.7.1. List nodes

Method	URI	Description
GET	/v1/nodes{?limit,marker,sort,global_project,cluster_id,name,status}	Lists all nodes.

Normal response codes: 200

3.7.1.1. Request

This table shows the query parameters for the list nodes request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
sort	String (Optional)	Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is <code>asc</code> (ascending) or <code>desc</code> (descending). Default direction is <code>asc</code> (ascending). Specify the list as <code><key>[:<direction>]</code> . For example, the following query parameters in the URI sort the objects in the response by name in ascending order and then by status in descending order: <code>GET /v2/images?sort=name:asc,status:desc</code> The following query parameters in the URI sort the objects in the response by name in descending order and then by status in ascending order. <code>GET /v2/images?sort=name:desc,status</code>
global_project	Boolean (Optional)	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to <code>true</code> , the call returns all objects from all projects. Default is <code>false</code> , which returns only objects in the current project.
cluster_id	String (Optional)	Filters the response by the cluster that owns a node.
name	String (Optional)	Filters the response by the name of a node.
status	String (Optional)	Filters the response by the status of a node.

This operation does not accept a request body.

3.7.1.2. Response

Example 3.15. List nodes: JSON response

```
{
  "nodes": [
    {
      "cluster_id": null,
      "created_at": "2015-02-27T04:39:21",
      "data": {},
      "details": {},
      "domain": null,
      "id": "573aa1ba-bf45-49fd-907d-6b5d6e6adfd3",
      "index": -1,
      "init_at": "2015-02-27T04:39:18",
      "metadata": {},
      "name": "node00a",
      "physical_id": "cc028275-d078-4729-bf3e-154b7359814b",
      "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
      "profile_name": "mystack",
      "project": "6e18cc2bdb48a5b3cad2dc499f6804",
      "role": null,
      "status": "ACTIVE",
      "status_reason": "Creation succeeded",
      "updated_at": null,
      "user": "5e5bf8027826429c96af157f68dc9072"
    }
  ]
}
```

3.7.2. Create node

Method	URI	Description
POST	/v1/nodes	Creates a node.

Normal response codes: 202

3.7.2.1. Request

Example 3.16. Create node: JSON request

```
{
  "node": {
    "cluster_id": null,
    "metadata": {},
    "name": "node009",
    "profile_id": "mystack",
    "role": "master"
  }
}
```

3.7.2.2. Response

Example 3.17. Create node: JSON response

```
{
  "node": {
    "action": "2366d440-c73e-4961-9254-6d1c3af7c167",
    "cluster_id": null,
    "created_at": null,
    "data": {},
    "domain": null,
    "id": "0df0931b-e251-4f2e-8719-4ebfda3627ba",
    "index": -1,
    "init_at": "2015-03-05T08:53:15",
    "metadata": {},
    "name": "node009",
    "physical_id": "",
    "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
    "profile_name": "mystack",
    "project": "6e18cc2bdb48a5b3cad2dc499f6804",
    "role": "master",
    "status": "INIT",
    "status_reason": "Initializing",
    "updated_at": null,
    "user": "5e5bf8027826429c96af157f68dc9072"
  }
}
```

3.7.3. Show node details

Method	URI	Description
GET	/v1/nodes/{node_id}{?show_details}	Shows data for a node.

Normal response codes: 200

3.7.3.1. Request

This table shows the URI parameters for the show node details request:

Name	Type	Description
{node_id}	UUID	The UUID of the node.

This table shows the query parameters for the show node details request:

Name	Type	Description
show_details	Boolean (Optional)	Indicates whether the node details are returned. Default is false.

This operation does not accept a request body.

3.7.3.2. Response

Example 3.18. Show node details: JSON response

A node record. Each record contains fields such as `id`, `cluster_id`, `name`, `physical_id`, `profile_id`, `created_at`, `index`, `status`, `status_reason`, `metadata`, `updated_at`, and so on.

```
{
  "node": {
    "cluster_id": null,
    "created_at": "2015-02-10T12:03:16",
    "data": {},
    "domain": null,
    "id": "d5779bb0-f0a0-49c9-88cc-6f078adb5a0b",
    "index": -1,
    "init_at": "2015-02-10T12:03:13",
    "metadata": {},
    "name": "node1",
    "physical_id": "f41537fa-22ab-4bea-94c0-c874e19d0c80",
    "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
    "profile_name": "mystack",
    "project": "6e18cc2bdb48a5b3cad2dc499f6804",
    "role": null,
    "status": "ACTIVE",
    "status_reason": "Creation succeeded",
    "updated_at": "2015-03-04T04:58:27",
    "user": "5e5bf8027826429c96af157f68dc9072"
  }
}
```

3.7.4. Update node

Method	URI	Description
PATCH	/v1/nodes/{node_id}	Updates a node.

Normal response codes: 202

3.7.4.1. Request

This table shows the URI parameters for the update node request:

Name	Type	Description
{node_id}	UUID	The UUID of the node.

Example 3.19. Update node: JSON request

```
{
  "node": {
    "name": "new_node_name"
  }
}
```

3.7.4.2. Response

Example 3.20. Update node: JSON response

```
{
  "node": {
    "action": "2366d440-c73e-4961-9254-6d1c3af7c167",
    "cluster_id": null,
    "created_at": null,
    "data": {},
    "domain": null,
    "id": "0df0931b-e251-4f2e-8719-4ebfda3627ba",
    "index": -1,
    "init_at": "2015-03-05T08:53:15",
    "metadata": {},
    "name": "node009",
    "physical_id": "",
    "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
    "profile_name": "mystack",
    "project": "6e18cc2bdb48a5b3cad2dc499f6804",
    "role": "master",
    "status": "ACTIVE",
    "status_reason": "Update succeeded",
    "updated_at": "2015-09-01T18:53:15",
    "user": "5e5bf8027826429c96af157f68dc9072"
  }
}
```

3.7.5. Delete node

Method	URI	Description
DELETE	/v1/nodes/{node_id}	Deletes a node.

Normal response codes: 202

3.7.5.1. Request

This table shows the URI parameters for the delete node request:

Name	Type	Description
{node_id}	UUID	The UUID of the node.

This operation does not accept a request body.

3.7.5.2. Response

The Location response header contains the action that the request initiated.

This operation does not return a response body.

3.7.6. Trigger node action

Method	URI	Description
POST	/v1/nodes/{node_id}/actions	Triggers an action on a node.

Normal response codes: 202

3.7.6.1. Request

This table shows the URI parameters for the trigger node action request:

Name	Type	Description
{node_id}	UUID	The UUID of the node.

3.7.6.2. Response

The Location response header contains the action that the request initiates.

This operation does not return a response body.

3.8. Policies

Lists all policies and creates, shows information for, updates, and deletes a policy.

Method	URI	Description
GET	/v1/policies{?limit,marker,sort,global_project,name,type}	Lists all policies.
POST	/v1/policies	Creates a policy.
GET	/v1/policies/{policy_id}	Shows details for a policy.
PATCH	/v1/policies/{policy_id}	Updates a policy.
DELETE	/v1/policies/{policy_id}	Deletes a policy.

3.8.1. List policies

Method	URI	Description
GET	/v1/policies{?limit,marker,sort,global_project,name,type}	Lists all policies.

Normal response codes: 200

3.8.1.1. Request

This table shows the query parameters for the list policies request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
sort	String (Optional)	Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is <code>asc</code> (ascending) or <code>desc</code> (descending). Default direction is <code>asc</code> (ascending). Specify the list as <code><key>[:<direction>]</code> . For example, the following query parameters in the URI sort the objects in the response by name in ascending order and then by status in descending order: <code>GET /v2/images?sort=name:asc,status:desc</code> The following query parameters in the URI sort the objects in the response by name in descending order and then by status in ascending order. <code>GET /v2/images?sort=name:desc,status</code>
global_project	Boolean (Optional)	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to <code>true</code> , the call returns all objects from all projects. Default is <code>false</code> , which returns only objects in the current project.
name	String (Optional)	Filters the response by the name of a policy.
type	String (Optional)	Filters the response by the type of a policy.

This operation does not accept a request body.

3.8.1.2. Response

Example 3.21. List policies: JSON response

```
{  
  "policies": [  

```

```
{
  "created_at": "2015-02-15T08:33:13.000000",
  "data": {},
  "domain": null,
  "id": "7192d8df-73be-4e98-ab99-1cf6d5066729",
  "name": "test_policy_1",
  "project": "42d9e9663331431f97b75e25136307ff",
  "spec": {
    "description": "A test policy",
    "properties": {
      "criteria": "OLDEST_FIRST",
      "destroy_after_deletion": true,
      "grace_period": 60,
      "reduce_desired_capacity": false
    },
    "type": "senlin.policy.deletion",
    "version": "1.0"
  },
  "type": "senlin.policy.deletion-1.0",
  "updated_at": null,
  "user": "5e5bf8027826429c96af157f68dc9072"
}
```

3.8.2. Create policy

Method	URI	Description
POST	/v1/policies	Creates a policy.

Normal response codes: 201

3.8.2.1. Request

Example 3.22. Create policy: JSON request

```
{
  "policy": {
    "name": "sp001",
    "spec": {
      "properties": {
        "adjustment": {
          "min_step": 1,
          "number": 1,
          "type": "CHANGE_IN_CAPACITY"
        },
        "event": "CLUSTER_SCALE_IN"
      },
      "type": "senlin.policy.scaling",
      "version": "1.0"
    }
  }
}
```

3.8.2.2. Response

Example 3.23. Create policy: JSON response

```
{
  "policy": {
    "created_at": "2015-03-02T07:40:31",
    "data": {},
    "domain": null,
    "id": "02f62195-2198-4797-b0a9-877632208527",
    "name": "sp001",
    "project": "42d9e9663331431f97b75e25136307ff",
    "spec": {
      "properties": {
        "adjustment": {
          "best_effort": true,
          "min_step": 1,
          "number": 1,
          "type": "CHANGE_IN_CAPACITY"
        },
        "event": "CLUSTER_SCALE_IN"
      },
      "type": "senlin.policy.scaling",
      "version": "1.0"
    },
    "type": "senlin.policy.scaling-1.0",
    "updated_at": null,
  }
}
```

```
    "user": "5e5bf8027826429c96af157f68dc9072"  
  }  
}
```

3.8.3. Show policy details

Method	URI	Description
GET	/v1/policies/{policy_id}	Shows details for a policy.

Normal response codes: 200

3.8.3.1. Request

This table shows the URI parameters for the show policy details request:

Name	Type	Description
{policy_id}	UUID	The UUID of the policy.

This operation does not accept a request body.

3.8.3.2. Response

Example 3.24. Show policy details: JSON response

A policy record with fields such as `id`, `name`, `type`, `spec`, `created_at`, `updated_at`, and so on.

```
{
  "policy": {
    "created_at": "2015-03-02T07:40:31",
    "data": {},
    "domain": null,
    "id": "02f62195-2198-4797-b0a9-877632208527",
    "name": "sp001",
    "project": "42d9e9663331431f97b75e25136307ff",
    "spec": {
      "properties": {
        "adjustment": {
          "best_effort": true,
          "min_step": 1,
          "number": 1,
          "type": "CHANGE_IN_CAPACITY"
        },
        "event": "CLUSTER_SCALE_IN"
      },
      "type": "senlin.policy.scaling",
      "version": "1.0"
    },
    "type": "senlin.policy.scaling-1.0",
    "updated_at": null,
    "user": "5e5bf8027826429c96af157f68dc9072"
  }
}
```

3.8.4. Update policy

Method	URI	Description
PATCH	/v1/policies/{policy_id}	Updates a policy.

Normal response codes: 200

3.8.4.1. Request

This table shows the URI parameters for the update policy request:

Name	Type	Description
{policy_id}	UUID	The UUID of the policy.

Example 3.25. Update policy: JSON request

```
{
  "policy": {
    "name": "new_name"
  }
}
```

3.8.4.2. Response

Example 3.26. Update policy: JSON response

A policy record that includes fields such as id, name, type, spec, created_at, updated_at, and so on.

```
{
  "policy": {
    "created_at": "2015-10-14T09:14:53",
    "data": {},
    "domain": null,
    "id": "ac5415bd-f522-4160-8be0-f8853e4bc332",
    "name": "dp01",
    "project": "42d9e9663331431f97b75e25136307ff",
    "spec": {
      "description": "A policy for node deletion.",
      "properties": {
        "criteria": "OLDEST_FIRST",
        "destroy_after_deletion": true,
        "grace_period": 60,
        "reduce_desired_capacity": false
      },
      "type": "senlin.policy.deletion",
      "version": "1.0"
    },
    "type": "senlin.policy.deletion-1.0",
    "updated_at": null,
    "user": "5e5bf8027826429c96af157f68dc9072"
  }
}
```

3.8.5. Delete policy

Method	URI	Description
DELETE	/v1/policies/{policy_id}	Deletes a policy.

Normal response codes: 204

3.8.5.1. Request

This table shows the URI parameters for the delete policy request:

Name	Type	Description
{policy_id}	UUID	The UUID of the policy.

This operation does not accept a request body.

3.9. Policy types

Lists all policy types and shows details for a policy type.

Method	URI	Description
GET	/v1/policy-types	Lists all supported policy types.
GET	/v1/policy-types/{policy_type}	Shows details for a policy type.

3.9.1. List policy types

Method	URI	Description
GET	/v1/policy-types	Lists all supported policy types.

Normal response codes: 200

3.9.1.1. Request

This operation does not accept a request body.

3.9.1.2. Response

Example 3.27. List policy types: JSON response

A list of supported policy types.

```
{
  "policy_types": [
    {
      "name": "ScalingPolicy"
    },
    {
      "name": "PlacementPolicy"
    },
    {
      "name": "DeletionPolicy"
    },
    {
      "name": "LoadBalancingPolicy"
    },
    {
      "name": "HealthPolicy"
    },
    {
      "name": "UpdatePolicy"
    }
  ]
}
```


3.9.2. Show policy type details

Method	URI	Description
GET	/v1/policy-types/{policy_type}	Shows details for a policy type.

Normal response codes: 200

3.9.2.1. Request

This table shows the URI parameters for the show policy type details request:

Name	Type	Description
{policy_type}	String	The name of the policy type.

This operation does not accept a request body.

3.9.2.2. Response

Example 3.28. Show policy type details: JSON response

```
{
  "policy_type": {
    "name": "senlin.policy.deletion",
    "schema": {
      "criteria": {
        "constraints": [
          {
            "constraint": [
              "OLDEST_FIRST",
              "OLDEST_PROFILE_FRIST",
              "YOUNGEST_FIRST",
              "RANDOM"
            ],
            "type": "AllowedValues"
          }
        ],
        "default": "RANDOM",
        "description": "Criteria used in selecting candidates for
deletion",
        "required": false,
        "type": "String"
      },
      "destroy_after_deletion": {
        "default": true,
        "description": "Whether a node should be completely destroyed
after deletion. Default to true",
        "required": false,
        "type": "Boolean"
      },
      "grace_period": {
        "default": 0,
        "description": "Number of seconds before real deletion
happens.",
        "required": false,
        "type": "Integer"
      }
    }
  }
}
```

```
    },
    "reduce_desired_capacity": {
      "default": false,
      "description": "Whether the desired capacity of the cluster
should be reduced along the deletion. Default to false.",
      "required": false,
      "type": "Boolean"
    }
  }
}
```

3.10. Profiles

Lists all profiles and creates, shows information for, updates, and deletes a profile.

Method	URI	Description
GET	/v1/profiles{?limit,marker,sort,global_project,name,type,metadata}	Lists all profiles.
POST	/v1/profiles	Creates a profile.
GET	/v1/profiles/{profile_id}	Shows details for a profile.
PATCH	/v1/profiles/{profile_id}	Updates a profile.
DELETE	/v1/profiles/{profile_id}	Deletes a profile.

3.10.1. List profiles

Method	URI	Description
GET	/v1/profiles{?limit,marker,sort,global_project,name,type,metadata}	Lists all profiles.

Normal response codes: 200

3.10.1.1. Request

This table shows the query parameters for the list profiles request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
sort	String (Optional)	Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is <code>asc</code> (ascending) or <code>desc</code> (descending). Default direction is <code>asc</code> (ascending). Specify the list as <code><key>[:<direction>]</code> . For example, the following query parameters in the URI sort the objects in the response by name in ascending order and then by status in descending order: <code>GET /v2/images?sort=name:asc,status:desc</code> The following query parameters in the URI sort the objects in the response by name in descending order and then by status in ascending order. <code>GET /v2/images?sort=name:desc,status</code>
global_project	Boolean (Optional)	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to <code>true</code> , the call returns all objects from all projects. Default is <code>false</code> , which returns only objects in the current project.
name	String (Optional)	Filters the response by the name of a profile.
type	String (Optional)	Filters the response by the type of a profile.
metadata	Dict (Optional)	Filters the response by a metadata key and value pair.

This operation does not accept a request body.

3.10.1.2. Response

Example 3.29. List profiles: JSON response

```
{
```

```

"profiles": [
  {
    "created_at": "2015-02-10T11:46:33.000000",
    "domain": null,
    "id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
    "metadata": {},
    "name": "mystack",
    "project": "42d9e9663331431f97b75e25136307ff",
    "spec": {
      "properties": {
        "disable_rollback": false,
        "environment": {
          "resource_registry": {
            "os.heat.server": "OS::Heat::Server"
          }
        },
        "files": {
          "file:///opt/stack/senlin/examples/profiles/
test_script.sh": "#!/bin/bash\n\nnecho \"this is a test script file\"\n\n",
        },
        "name": "random_string_stack",
        "parameters": {},
        "rollback": false,
        "template": {
          "heat_template_version": "2014-10-16",
          "outputs": {
            "result": {
              "value": {
                "get_attr": [
                  "random",
                  "value"
                ]
              }
            }
          },
          "parameters": {
            "file": {
              "default": {
                "get_file": "file:///opt/stack/senlin/
examples/profiles/test_script.sh"
              },
              "type": "string"
            }
          },
          "resources": {
            "random": {
              "properties": {
                "length": 64
              },
              "type": "OS::Heat::RandomString"
            }
          }
        },
        "timeout": 60
      },
      "type": "os.heat.stack",
      "version": "1.0"
    },
    "type": "os.heat.stack-1.0",
    "updated_at": null,
  }
]

```

```
    "user": "5e5bf8027826429c96af157f68dc9072"  
  }  
]  
}
```

3.10.2. Create profile

Method	URI	Description
POST	/v1/profiles	Creates a profile.

Normal response codes: 201

3.10.2.1. Request

Example 3.30. Create profile: JSON request

```
{
  "profile": {
    "metadata": {},
    "name": "test_prof1",
    "spec": {
      "properties": {
        "disable_rollback": false,
        "environment": {
          "resource_registry": {
            "os.heat.server": "OS::Heat::Server"
          }
        },
        "files": {
          "file:///usr/test_script.sh": "#!/bin/bash\n\nnecho \"this
is a test script file\"\n"
        },
        "parameters": {},
        "template": {
          "heat_template_version": "2014-10-16",
          "outputs": {
            "result": {
              "value": {
                "get_attr": [
                  "random",
                  "value"
                ]
              }
            }
          }
        },
        "parameters": {
          "file": {
            "default": {
              "get_file": "file:///usr/test_script.sh"
            },
            "type": "string"
          }
        },
        "resources": {
          "random": {
            "properties": {
              "length": 64
            },
            "type": "OS::Heat::RandomString"
          }
        }
      }
    }
  },
}
```

```

        "timeout": 60
      },
      "type": "os.heat.stack",
      "version": "1.0"
    }
  }
}

```

3.10.2.2. Response

Example 3.31. Create profile: JSON response

```

{
  "profile": {
    "created_at": "2015-03-01T14:28:25",
    "domain": null,
    "id": "7fa885cd-fa39-4531-a42d-780af95c84a4",
    "metadata": {},
    "name": "test_prof1",
    "project": "42d9e9663331431f97b75e25136307ff",
    "spec": {
      "properties": {
        "disable_rollback": false,
        "environment": {
          "resource_registry": {
            "os.heat.server": "OS::Heat::Server"
          }
        },
        "files": {
          "file:///opt/stack/senlin/examples/profiles/test_script.sh": "#!/bin/bash\n\nnecho \"this is a test script file\"\n\n"
        },
        "parameters": {},
        "template": {
          "heat_template_version": "2014-10-16",
          "outputs": {
            "result": {
              "value": {
                "get_attr": [
                  "random",
                  "value"
                ]
              }
            }
          }
        },
        "parameters": {
          "file": {
            "default": {
              "get_file": "file:///opt/stack/senlin/examples/profiles/test_script.sh"
            },
            "type": "string"
          }
        },
        "resources": {
          "random": {
            "properties": {
              "length": 64
            },
            "type": "OS::Heat::RandomString"
          }
        }
      }
    }
  }
}

```

```
        }
      },
      "timeout": 60
    },
    "type": "os.heat.stack",
    "version": "1.0"
  },
  "type": "os.heat.stack-1.0",
  "updated_at": null,
  "user": "5e5bf8027826429c96af157f68dc9072"
}
```


3.10.3. Show profile details

Method	URI	Description
GET	/v1/profiles/{profile_id}	Shows details for a profile.

Normal response codes: 200

3.10.3.1. Request

This table shows the URI parameters for the show profile details request:

Name	Type	Description
{profile_id}	UUID	The UUID of the profile.

This operation does not accept a request body.

3.10.3.2. Response

Example 3.32. Show profile details: JSON response

A profile record. Contains fields such as id, name, type, spec, metadata, and so on.

```
{
  "profile": {
    "created_at": "2015-03-01T14:28:25",
    "domain": null,
    "id": "7fa885cd-fa39-4531-a42d-780af95c84a4",
    "metadata": {},
    "name": "test_prof1",
    "project": "42d9e9663331431f97b75e25136307ff",
    "spec": {
      "disable_rollback": false,
      "environment": {
        "resource_registry": {
          "os.heat.server": "OS::Heat::Server"
        }
      },
      "files": {
        "file:///opt/stack/senlin/examples/profiles/test_script.sh":
        "#!/bin/bash\nnecho \"this is a test script file\"\n"
      },
      "parameters": {},
      "template": {
        "heat_template_version": "2014-10-16",
        "outputs": {
          "result": {
            "value": {
              "get_attr": [
                "random",
                "value"
              ]
            }
          }
        }
      }
    },
    "parameters": {
```

```
        "file": {
            "default": {
                "get_file": "file:///opt/stack/senlin/examples/
profiles/test_script.sh"
            },
            "type": "string"
        },
        "resources": {
            "random": {
                "properties": {
                    "length": 64
                },
                "type": "OS::Heat::RandomString"
            },
            "timeout": 60
        },
        "type": "os.heat.stack",
        "version": "1.0"
    },
    "type": "os.heat.stack-1.0",
    "updated_at": null,
    "user": "5e5bf8027826429c96af157f68dc9072"
}
```

3.10.4. Update profile

Method	URI	Description
PATCH	/v1/profiles/{profile_id}	Updates a profile.

Normal response codes: 200

3.10.4.1. Request

This table shows the URI parameters for the update profile request:

Name	Type	Description
{profile_id}	UUID	The UUID of the profile.

Example 3.33. Update profile: JSON request

```
{
  "profile": {
    "metadata": {},
    "name": null
  }
}
```

3.10.4.2. Response

Example 3.34. Update profile: JSON response

A profile record. Contains fields such as id, name, type, spec, metadata, and so on.

```
{
  "profile": {
    "created_at": "2015-03-01T14:28:25",
    "domain": null,
    "id": "7fa885cd-fa39-4531-a42d-780af95c84a4",
    "metadata": {},
    "name": "test_prof1",
    "project": "42d9e9663331431f97b75e25136307ff",
    "spec": {
      "disable_rollback": false,
      "environment": {
        "resource_registry": {
          "os.heat.server": "OS::Heat::Server"
        }
      },
      "files": {
        "file:///opt/stack/senlin/examples/profiles/test_script.sh":
        "#!/bin/bash\nnecho \"this is a test script file\"\n"
      },
      "parameters": {},
      "template": {
        "heat_template_version": "2014-10-16",
        "outputs": {
          "result": {
            "value": {
              "get_attr": [
```

```
        "random",
        "value"
      ]
    }
  },
  "parameters": {
    "file": {
      "default": {
        "get_file": "file:///opt/stack/senlin/examples/
profiles/test_script.sh"
      },
      "type": "string"
    }
  },
  "resources": {
    "random": {
      "properties": {
        "length": 64
      },
      "type": "OS::Heat::RandomString"
    }
  },
  "timeout": 60
},
"type": "os.heat.stack",
"version": "1.0"
},
"type": "os.heat.stack-1.0",
"updated_at": null,
"user": "5e5bf8027826429c96af157f68dc9072"
}
}
```

3.10.5. Delete profile

Method	URI	Description
DELETE	/v1/profiles/{profile_id}	Deletes a profile.

Normal response codes: 204

3.10.5.1. Request

This table shows the URI parameters for the delete profile request:

Name	Type	Description
{profile_id}	UUID	The UUID of the profile.

This operation does not accept a request body.

3.11. Profile types

Lists all profile types and shows details for a profile type.

Method	URI	Description
GET	/v1/profile-types	Lists supported profile types.
GET	/v1/profile-types/{profile_type}	Shows details for a profile type.

3.11.1. List profile types

Method	URI	Description
GET	/v1/profile-types	Lists supported profile types.

Normal response codes: 200

3.11.1.1. Request

This operation does not accept a request body.

3.11.1.2. Response

Example 3.35. List profile types: JSON response

A list of supported profile types.

```
{
  "profile_types": [
    {
      "name": "os.heat.stack"
    },
    {
      "name": "os.heat.resource"
    },
    {
      "name": "os.nova.server"
    }
  ]
}
```

3.11.2. Show profile type details

Method	URI	Description
GET	/v1/profile-types/{profile_type}	Shows details for a profile type.

Normal response codes: 200

3.11.2.1. Request

This table shows the URI parameters for the show profile type details request:

Name	Type	Description
{profile_type}	String	The name of the profile type.

This operation does not accept a request body.

3.11.2.2. Response

Example 3.36. Show profile type details: JSON response

```
{
  "profile_type": {
    "name": "os.heat.stack",
    "schema": {
      "context": {
        "default": {},
        "description": "A dictionary for specifying the customized
context for stack operations",
        "required": false,
        "type": "Map"
      },
      "disable_rollback": {
        "default": true,
        "description": "A boolean specifying whether a stack operation
can be rolled back.",
        "required": false,
        "type": "Boolean"
      },
      "environment": {
        "default": {},
        "description": "A map that specifies the environment used for
stack operations.",
        "required": false,
        "type": "Map"
      },
      "files": {
        "default": {},
        "description": "Contents of files referenced by the template,
if any.",
        "required": false,
        "type": "Map"
      },
      "parameters": {
        "default": {},
        "description": "Parameters to be passed to Heat for stack
operations.",

```

```
        "required": false,
        "type": "Map"
      },
      "template": {
        "description": "Heat stack template.",
        "required": true,
        "type": "Map"
      },
      "timeout": {
        "description": "A integer that specifies the number of minutes
that a stack operation times out.",
        "required": false,
        "type": "Integer"
      }
    }
  }
}
```

3.12. Receivers

Lists all receivers and creates, shows information for, and deletes a receiver.

Method	URI	Description
GET	/v1/receivers{?limit,marker,sort,global_project,name,type,user,cluster_id,action}	Lists all receivers.
POST	/v1/receivers	Creates a receiver.
GET	/v1/receivers/{receiver_id}	Shows details for a receiver.
DELETE	/v1/receivers/{receiver_id}	Deletes a receiver.

3.12.1. List receivers

Method	URI	Description
GET	/v1/receivers{?limit,marker,sort,global_project,name,type,user,cluster_id,action}	Lists all receivers.

Normal response codes: 200

3.12.1.1. Request

This table shows the query parameters for the list receivers request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
sort	String (Optional)	Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is <code>asc</code> (ascending) or <code>desc</code> (descending). Default direction is <code>asc</code> (ascending). Specify the list as <code><key>[:<direction>]</code> . For example, the following query parameters in the URI sort the objects in the response by name in ascending order and then by status in descending order: <code>GET /v2/images?sort=name:asc,status:desc</code> The following query parameters in the URI sort the objects in the response by name in descending order and then by status in ascending order. <code>GET /v2/images?sort=name:desc,status</code>
global_project	Boolean (Optional)	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to <code>true</code> , the call returns all objects from all projects. Default is <code>false</code> , which returns only objects in the current project.
name	String (Optional)	Filters the response by the name of a receiver.
type	String (Optional)	Filters the response by the type of a receiver.
user	String (Optional)	Filters the response by the user name of a receiver.
cluster_id	String (Optional)	Filters the response by the ID of the targeted cluster of a receiver.
action	String (Optional)	Filters the response by the targeted action of a receiver.

This operation does not accept a request body.

3.12.1.2. Response

Example 3.37. List receivers: JSON response

List of receiver records. Each record contains fields such as `id`, `name`, `user`, `project`, `domain`, `cluster_id`, `type`, `action`, `actor`, `params`, `created_at` and `updated_at`, and so on.

```
{
  "receivers": [
    {
      "action": "CLUSTER_SCALE_OUT",
      "actor": {
        "trust_id": [
          "6dc6d336e3fc4c0a951b5698cd1236d9"
        ]
      },
      "channel": {
        "alarm_url": "http://node1:8778/v1/webhooks/
e03dd2e5-8f2e-4ec1-8c6a-74ba891e5422/trigger?V=1&count=1"
      },
      "cluster_id": "ae63a10b-4a90-452c-aef1-113a0b255ee3",
      "created_at": "2015-06-27T05:09:43",
      "domain": "Default",
      "id": "573aa1ba-bf45-49fd-907d-6b5d6e6adfd3",
      "name": "cluster_inflate",
      "params": {
        "count": "1"
      },
      "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
      "type": "webhook",
      "updated_at": null,
      "user": "b4ad2d6e18cc2b9c48049f6dbe8a5b3c"
    }
  ]
}
```

3.12.2. Create receiver

Method	URI	Description
POST	/v1/receivers	Creates a receiver.

Normal response codes: 201

3.12.2.1. Request

Example 3.38. Create receiver: JSON request

```
{
  "receiver": {
    "action": "CLUSTER_SCALE_OUT",
    "cluster_id": "cf99d754-3cdc-47f4-8a29-cd14f02f5436",
    "name": "cluster_inflate",
    "params": {
      "count": "1"
    },
    "type": "webhook"
  }
}
```

3.12.2.2. Response

Example 3.39. Create receiver: JSON response

```
{
  "receiver": {
    "action": "CLUSTER_SCALE_OUT",
    "actor": {
      "trust_id": [
        "6dc6d336e3fc4c0a951b5698cd1236d9"
      ]
    },
    "channel": {
      "alarm_url": "http://node1:8778/v1/webhooks/e03dd2e5-8f2e-4ec1-8c6a-74ba891e5422/trigger?V=1&count=1"
    },
    "cluster_id": "ae63a10b-4a90-452c-aef1-113a0b255ee3",
    "created_at": "2015-06-27T05:09:43",
    "domain": "Default",
    "id": "573aa1ba-bf45-49fd-907d-6b5d6e6adfd3",
    "name": "cluster_inflate",
    "params": {
      "count": "1"
    },
    "project": "6e18cc2bdb48a5b3cad2dc499f6804",
    "type": "webhook",
    "updated_at": null,
    "user": "b4ad2d6e18cc2b9c48049f6dbe8a5b3c"
  }
}
```

3.12.3. Show receiver details

Method	URI	Description
GET	/v1/receivers/{receiver_id}	Shows details for a receiver.

Normal response codes: 200

3.12.3.1. Request

This table shows the URI parameters for the show receiver details request:

Name	Type	Description
{receiver_id}	UUID	The UUID of the receiver.

This operation does not accept a request body.

3.12.3.2. Response

Example 3.40. Show receiver details: JSON response

A receiver record. The record includes fields such as `id`, `name`, `type`, `cluster_id`, `action`, `actor`, `params`, `channel`, `created_at`, `updated_at`, `user`, `project`, `domain`, and so on.

```
{
  "receiver": {
    "action": "CLUSTER_SCALE_OUT",
    "actor": {
      "trust_id": [
        "6dc6d336e3fc4c0a951b5698cd1236d9"
      ]
    },
    "channel": {
      "alarm_url": "http://node1:8778/v1/webhooks/e03dd2e5-8f2e-4ec1-8c6a-74ba891e5422/trigger?V=1&count=1"
    },
    "cluster_id": "ae63a10b-4a90-452c-aef1-113a0b255ee3",
    "created_at": "2015-06-27T05:09:43",
    "domain": "Default",
    "id": "573aa1ba-bf45-49fd-907d-6b5d6e6adfd3",
    "name": "cluster_inflate",
    "params": {
      "count": "1"
    },
    "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
    "type": "webhook",
    "updated_at": null,
    "user": "b4ad2d6e18cc2b9c48049f6dbe8a5b3c"
  }
}
```

3.12.4. Delete receiver

Method	URI	Description
DELETE	/v1/receivers/{receiver_id}	Deletes a receiver.

Normal response codes: 204

3.12.4.1. Request

This table shows the URI parameters for the delete receiver request:

Name	Type	Description
{receiver_id}	UUID	The UUID of the receiver.

This operation does not accept a request body.

3.13. Webhooks

Triggers an action represented by a webhook.

Method	URI	Description
POST	/v1/webhooks/{webhook_id}/trigger {?V,params}	Triggers a webhook receiver.

3.13.1. Trigger webhook action

Method	URI	Description
POST	/v1/webhooks/{webhook_id}/trigger {?V,params}	Triggers a webhook receiver.

Normal response codes: 202

3.13.1.1. Request

This table shows the URI parameters for the trigger webhook action request:

Name	Type	Description
{webhook_id}	UUID	The UUID of the webhook.

This table shows the query parameters for the trigger webhook action request:

Name	Type	Description
V	String (Required)	The webhook implementation version requested.
params	Dict (Optional)	The query string that forms the inputs to use for the targeted action.

This operation does not accept a request body.

3.13.1.2. Response

The Location response header contains the action that the request initiated.

This operation does not return a response body.

4. Compute API (CURRENT)

This documentation supports the API v2.1 base version. If you enable all extensions, the Compute API v1.1 and v2.0 requests and responses match the API v2.1 requests and responses.

Each API version lists the extensions that it requires to make the full request and responses available. API v2.1 must enable all extensions all the time. It uses micro-version headers to expose any additional functionality.



Note

Effective in the OpenStack Kilo release, the Compute API no longer supports XML in requests and responses.

For details about the Compute API, see the [Compute API](#).

Method	URI	Description
API versions		
GET	/	Lists information about all Compute API versions.
GET	/v2.1	Show information about Compute API v2.1.
GET	/v2	Show information about Compute API v2.
Servers (servers)		
GET	/v2.1/{tenant_id}/servers{?changes-since,image,flavor,name,status,host,limit,marker}	Lists IDs, names, and links for all servers.
POST	/v2.1/{tenant_id}/servers	Creates a server.
GET	/v2.1/{tenant_id}/servers/detail{?changes-since,image,flavor,name,status,host,limit,marker}	Lists all servers with details.
GET	/v2.1/{tenant_id}/servers/{server_id}	Shows details for a server.
PUT	/v2.1/{tenant_id}/servers/{server_id}	Updates the editable attributes of a server.
DELETE	/v2.1/{tenant_id}/servers/{server_id}	Deletes a server.
Servers multiple create (servers)		
POST	/v2.1/{tenant_id}/servers	Creates one or more servers.
POST	/v2.1/{tenant_id}/servers	Creates one or more servers with a reservation ID.
Servers - run an action (servers, action)		
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a fixed IP address to a server instance, which associates that address with the server. The fixed IP address is retrieved from the network that you specify in the request.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to a server, which associates that address with the server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Changes the administrative password for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Confirms a pending resize action for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Creates an image from a server.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Evacuates a server from a failed host to a new one.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Force-deletes a server before deferred cleanup.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Locks a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Pauses a server. Changes its status to PAUSED.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Shows console output for a server instance.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets an RDP console for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a serial console for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a SPICE console for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a VNC console for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Reboots a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Rebuilds a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Removes, or disassociates, a fixed IP address from a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Removes, or disassociates, a floating IP address from a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Puts a server in rescue mode and changes its status to RESCUE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resizes a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Restores a previously soft-deleted server instance. You cannot use this method to restore deleted instances.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resumes a suspended server and changes its status to ACTIVE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Cancels and reverts a pending resize action for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Shelves a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Shelf-offloads, or removes, a shelved server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Starts a stopped server and changes its status to ACTIVE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Stops a running server and changes its status to SHUTOFF.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unlocks a locked server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unpauses a paused server and changes its status to ACTIVE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unrescues a server. Changes status to ACTIVE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unshelves, or restores, a shelved server.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a security group to a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Removes a security group from a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Trigger a crash dump in a server.
Servers - run an administrative action (servers, action)		
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Creates a back up of a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Injects network information into a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Migrates a server to a host. The scheduler chooses the host.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Live-migrates a server to a new host without rebooting.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resets networking on a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resets the state of a server.
Servers diagnostics (servers, diagnostics)		
GET	/v2.1/{tenant_id}/servers/{server_id}/diagnostics	Shows basic usage data for a server.
Servers IPs (servers, ips)		
GET	/v2.1/{tenant_id}/servers/{server_id}/ips	Lists IP addresses that are assigned to an instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/ips/{network_label}	Shows IP addresses details for a network label of a server instance.
Server metadata (servers, metadata)		
GET	/v2.1/{tenant_id}/servers/{server_id}/metadata	Lists all metadata for a server.
PUT	/v2.1/{tenant_id}/servers/{server_id}/metadata	Creates or replaces one or more metadata items for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/metadata	Updates one or more metadata items for a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}	Shows details for a metadata item, by key, for a server.
PUT	/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}	Creates or replaces a metadata item, by key, for a server.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}	Deletes a metadata item, by key, from a server.
Servers - list actions (servers, os-instance-actions)		
GET	/v2.1/{tenant_id}/servers/{server_id}/os-instance-actions	Lists actions for a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-instance-actions/{request_id}	Shows details for a server action.
Port interfaces (servers, os-interface)		
POST	/v2.1/{tenant_id}/servers/{server_id}/os-interface	Creates a port interface and uses it to attach a port to a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-interface	Lists port interfaces that are attached to a server.

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-interface/{port_id}	Shows details for a port interface that is attached to a server.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-interface/{port_id}	Detaches a port interface.
Servers password (servers, os-server-password)		
GET	/v2.1/{tenant_id}/servers/{server_id}/os-server-password	Shows the administrative password for a server.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-server-password	Clears the encrypted administrative password for a server, which removes it from the metadata server.
Servers virtual interfaces (servers, os-virtual-interfaces)		
GET	/v2.1/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Lists the virtual interfaces for an instance.
Servers with volume attachments (servers, os-volume_attachments)		
POST	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments	Attaches a volume to a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments	Lists the volume attachments for a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Shows details for a volume attachment.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Detaches a volume from a server.
Flavors with extended attributes (flavors)		
GET	/v2.1/{tenant_id}/flavors{?minDisk,minRam,sort_key,sort_dir,limit,marker}	Lists flavors.
POST	/v2.1/{tenant_id}/flavors	Creates a flavor.
GET	/v2.1/{tenant_id}/flavors/{flavor_id}	Shows details for a flavor.
DELETE	/v2.1/{tenant_id}/flavors/{flavor_id}	Deletes a flavor.
GET	/v2.1/{tenant_id}/flavors/detail{?minDisk,minRam,sort_key,sort_dir,limit,marker}	Lists flavors with details.
Flavors access (flavors, os-flavor-access, action)		
GET	/v2.1/{tenant_id}/flavors/os-flavor-access	Lists flavor access information.
POST	/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action	Adds flavor access to a tenant and flavor.
POST	/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action	Removes flavor access from a tenant and flavor.
Flavors extra-specs (flavors, os-flavor-extra-specs)		
POST	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Creates extra specs for a flavor, by ID.
GET	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Lists all extra specs for a flavor, by ID.
GET	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}	Shows an extra spec, by key, for a flavor, by ID.
PUT	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}	Updates an extra spec, by key, for a flavor, by ID.

Method	URI	Description
DELETE	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}	Deletes an extra spec, by key, for a flavor, by ID.
Keypairs (keypairs)		
GET	/v2.1/{tenant_id}/os-keypairs	Lists keypairs that are associated with the account.
POST	/v2.1/{tenant_id}/os-keypairs	Generates or imports a keypair.
DELETE	/v2.1/{tenant_id}/os-keypairs/{keypair_name}	Deletes a keypair.
GET	/v2.1/{tenant_id}/os-keypairs/{keypair_name}	Shows details for a keypair that is associated with the account.
Limits (limits)		
GET	/v2.1/{tenant_id}/limits	Shows rate and absolute limits for the tenant.
Extensions (extensions)		
GET	/v2.1/{tenant_id}/extensions	Lists available extensions.
GET	/v2.1/{tenant_id}/extensions/{alias}	Shows details for an extension, by alias.
Images		
GET	/v2.1/{tenant_id}/images{?changes-since,server,name,status,minDisk,minRam,type,limit,marker}	Lists IDs, names, and links for available images.
GET	/v2.1/{tenant_id}/images/detail{?changes-since,server,name,status,minDisk,minRam,type,limit,marker}	Lists all details for available images.
GET	/v2.1/{tenant_id}/images/{image_id}	Shows details for an image.
DELETE	/v2.1/{tenant_id}/images/{image_id}	Deletes an image.
Image metadata		
GET	/v2.1/{tenant_id}/images/{image_id}/metadata	Shows metadata for an image.
PUT	/v2.1/{tenant_id}/images/{image_id}/metadata	Creates or replaces metadata for an image.
POST	/v2.1/{tenant_id}/images/{image_id}/metadata	Updates metadata items, by key, for an image.
GET	/v2.1/{tenant_id}/images/{image_id}/metadata/{key}	Shows details for a metadata item, by key, for an image.
PUT	/v2.1/{tenant_id}/images/{image_id}/metadata/{key}	Creates or updates a metadata item, by key, for an image.
DELETE	/v2.1/{tenant_id}/images/{image_id}/metadata/{key}	Deletes a metadata item, by key, for an image.
Guest agents (os-agents)		
POST	/v2.1/{tenant_id}/os-agents	Creates an agent build.
GET	/v2.1/{tenant_id}/os-agents	Lists agent builds.
DELETE	/v2.1/{tenant_id}/os-agents	Deletes an existing agent build.
PUT	/v2.1/{tenant_id}/os-agents/{agent_build_id}	Updates an agent build.
Host aggregates (os-aggregates, action)		
POST	/v2.1/{tenant_id}/os-aggregates	Creates an aggregate in an availability zone.
GET	/v2.1/{tenant_id}/os-aggregates	Lists all aggregates. Includes the ID, name, and availability zone for each aggregate.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Shows details for an aggregate. Details include hosts and metadata.
DELETE	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Deletes an aggregate.
PUT	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Updates either or both the name and availability zone for an aggregate.
POST	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action	Adds a host to an aggregate.
POST	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action	Creates or replaces metadata for an aggregate.
POST	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action	Removes a host from an aggregate.
Assisted volume snapshots (os-assisted-volume-snapshots)		
POST	/v2.1/{tenant_id}/os-assisted-volume-snapshots	Creates an assisted volume snapshot.
DELETE	/v2.1/{tenant_id}/os-assisted-volume-snapshots/{snapshot_id}{?delete_info}	Deletes an assisted volume snapshot.
Availability zones (os-availability-zone)		
GET	/v2.1/{tenant_id}/os-availability-zone	Gets availability zone information.
GET	/v2.1/{tenant_id}/os-availability-zone/detail	Gets detailed availability zone information.
Bare metal nodes (os-baremetal-nodes)		
POST	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes	Adds a bare metal node to a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes	Lists the bare metal nodes that are associated with a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action	Adds an interface to a bare metal node that is associated with a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action	Deletes an interface from a bare metal node that is associated with a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}	Shows details for a bare metal node.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}	Deletes a bare metal node from a server.
Cells (os-cells, capacities)		
GET	/v2.1/{tenant_id}/os-cells	Lists cells.
GET	/v2.1/{tenant_id}/os-cells	Lists cells with details.
GET	/v2.1/{tenant_id}/os-cells/{cell_id}	Shows data for a cell.
GET	/v2.1/{tenant_id}/os-cells/{cell_id}/capacities	Shows capacities for a cell.
Root certificates (os-certificates)		
POST	/v2.1/{tenant_id}/os-certificates	Creates a certificate.
GET	/v2.1/{tenant_id}/os-certificates/{certificate_id}	Shows details for a certificate.
Cloudpipe (os-cloudpipe)		

Method	URI	Description
GET	/v2.1/{tenant_id}/os-cloudpipe	Lists cloudpipes.
POST	/v2.1/{tenant_id}/os-cloudpipe	Creates a cloudpipe.
POST	/v2.1/{tenant_id}/os-cloud-pipe/configure-project	Updates the virtual private network (VPN) IP address and port for a cloudpipe instance.
Server consoles (servers, os-consoles, os-console-auth-token)		
POST	/v2.1/{tenant_id}/servers/{server_id}/consoles	Creates a console for a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/consoles	Lists all consoles for a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/consoles/{console_id}	Shows details for a console for a server instance.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/consoles/{console_id}	Deletes a console for a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-console-auth-token	Shows the authentication token for a console for a server instance.
Fixed IPs (os-fixed-ips)		
GET	/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}	Shows details for a fixed IP address.
POST	/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}/action	Reserves or releases a fixed IP.
Floating IP DNS records (os-floating-ip-dns)		
GET	/v2.1/{tenant_id}/os-floating-ip-dns	Lists registered DNS domains published by the DNS drivers.
PUT	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}	Creates or updates a DNS domain.
DELETE	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}	Deletes a DNS domain and all associated host entries.
PUT	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Creates or updates a DNS entry.
GET	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Finds a unique DNS entry for a domain and name.
DELETE	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Deletes a DNS entry.
GET	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}	Lists DNS entries for a domain and IP.
Floating IP pools (os-floating-ip-pools)		
GET	/v2.1/{tenant_id}/os-floating-ip-pools	Lists floating IP pools.
Floating IPs (os-floating-ips)		
GET	/v2.1/{tenant_id}/os-floating-ips	Lists floating IP addresses associated with the tenant or account.
POST	/v2.1/{tenant_id}/os-floating-ips	Creates, or allocates, a floating IP address for the current project. By default, the floating IP address is allocated from the public pool.
GET	/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}	Shows details for a floating IP address, by ID, that is associated with the tenant or account.
DELETE	/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}	Deletes, or deallocates, a floating IP address from the current project and returns it to the pool from which it was allocated.
Floating IPs bulk (os-floating-ips-bulk)		
GET	/v2.1/{tenant_id}/os-floating-ips-bulk	Lists all floating IPs.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-floating-ips-bulk	Bulk-creates floating IPs.
POST	/v2.1/{tenant_id}/os-floating-ips-bulk/delete	Bulk-deletes floating IPs.
GET	/v2.1/{tenant_id}/os-floating-ips-bulk/{host_name}	Lists all floating IPs for a host.
Ping instances (os-fping)		
GET	/v2.1/{tenant_id}/os-fping/{all_tenants,include,exclude}	Run the fping utility to ping instances and report which ones are alive.
GET	/v2.1/{tenant_id}/os-fping/{instance_id}	Run the fping utility to ping an instance and report whether it is alive.
Hosts (os-hosts)		
GET	/v2.1/{tenant_id}/os-hosts	Lists hosts.
PUT	/v2.1/{tenant_id}/os-hosts/{host_name}	Enables or puts a host in maintenance mode.
GET	/v2.1/{tenant_id}/os-hosts/{host_name}	Shows details for a host.
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/reboot	Reboots a host.
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/shutdown	Shuts down a host.
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/startup	Starts a host.
Hypervisors (os-hypervisors)		
GET	/v2.1/{tenant_id}/os-hypervisors	Lists hypervisors.
GET	/v2.1/{tenant_id}/os-hypervisors/statistics	Shows summary statistics for all hypervisors over all compute nodes.
GET	/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}	Shows details for a hypervisor.
GET	/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}/uptime	Shows the uptime for a hypervisor.
Instance usage audit log (os-instance-usage-audit-log)		
GET	/v2.1/{tenant_id}/os-instance-usage-audit-log	Lists usage audits for an instance.
GET	/v2.1/{tenant_id}/os-instance-usage-audit-log/{before_timestamp}{?before_timestamp}	Lists usage audits that occurred before a specified time.
Migrations (os-migrations)		
GET	/v2.1/{tenant_id}/os-migrations	Lists in-progress migrations.
Networks (os-networks)		
POST	/v2.1/{tenant_id}/os-networks	Creates a network.
GET	/v2.1/{tenant_id}/os-networks	Lists networks for the project.
POST	/v2.1/{tenant_id}/os-networks/add	Adds a network to a project.
GET	/v2.1/{tenant_id}/os-networks/{network_id}	Shows details for a network.
DELETE	/v2.1/{tenant_id}/os-networks/{network_id}	Deletes a network.
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Associates a network with a host.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Disassociates a host from a network.
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Disassociates a network from a project. You can then reuse the network.
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Disassociates a project from a network.
Quota class (os-quota-class-sets)		
GET	/v2.1/{tenant_id}/os-quota-class-sets/{class_id}	Shows the quota for a class.
PUT	/v2.1/{tenant_id}/os-quota-class-sets/{class_id}	Updates quota for a class.
Quota sets (os-quota-sets)		
PUT	/v2.1/{admin_tenant_id}/os-quota-sets/{tenant_id}{?user_id}	Update the quotas for a project or a project and a user.
DELETE	/v2.1/{admin_tenant_id}/os-quota-sets/{tenant_id}{?user_id}	Reverts the quotas to default values for a project or a project and a user.
GET	/v2.1/{admin_tenant_id}/os-quota-sets/{tenant_id}/defaults	Lists the default quotas for a project.
GET	/v2.1/{admin_tenant_id}/os-quota-sets/{tenant_id}/detail	Lists quotas with details for a project or a project and a user.
Security groups (os-security-groups)		
GET	/v2.1/{tenant_id}/os-security-groups	Lists security groups.
POST	/v2.1/{tenant_id}/os-security-groups	Creates a security group.
GET	/v2.1/{tenant_id}/os-security-groups/{security_group_id}	Shows details for a security group.
PUT	/v2.1/{tenant_id}/os-security-groups/{security_group_id}	Updates a security group.
DELETE	/v2.1/{tenant_id}/os-security-groups/{security_group_id}	Deletes a security group.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-security-groups	Lists security groups for a server.
Rules for default security group (os-security-group-default-rules)		
GET	/v2.1/{tenant_id}/os-security-group-default-rules	Lists default security group rules.
POST	/v2.1/{tenant_id}/os-security-group-default-rules	Creates a default security group rule.
GET	/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Shows details for a security group rule.
DELETE	/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Deletes a security group rule.
Rules for security group (os-security-group-rules)		
POST	/v2.1/{tenant_id}/os-security-group-rules	Creates a rule for a security group.
DELETE	/v2.1/{tenant_id}/os-security-group-rules/{security_group_rule_id}	Deletes a security group rule.
Create external events (os-server-external-events)		

Method	URI	Description
POST	/v2.1/{tenant_id}/os-server-external-events	Creates one or more external events, which the API dispatches to the instance.
Server groups (os-server-groups)		
GET	/v2.1/{tenant_id}/os-server-groups{?all_projects}	Lists all server groups for the tenant.
POST	/v2.1/{tenant_id}/os-server-groups	Creates a server group.
GET	/v2.1/{tenant_id}/os-server-groups/{server_group_id}	Shows details for a server group.
DELETE	/v2.1/{tenant_id}/os-server-groups/{server_group_id}	Deletes a server group.
Compute services (os-services)		
GET	/v2.1/{tenant_id}/os-services	Lists all running Compute services for a tenant.
PUT	/v2.1/{tenant_id}/os-services/enable	Enables scheduling for a Compute service.
PUT	/v2.1/{tenant_id}/os-services/disable	Disables scheduling for a Compute service.
PUT	/v2.1/{tenant_id}/os-services/disable-log-reason	Logs information to the Compute service table about why a Compute service was disabled.
DELETE	/v2.1/{tenant_id}/os-services/{service_id}	Deletes a Compute service.
Usage reports (os-simple-tenant-usage)		
GET	/v2.1/{tenant_id}/os-simple-tenant-usage	Lists usage information for all tenants.
GET	/v2.1/{tenant_id}/os-simple-tenant-usage/{tenant_id}	Shows usage details for a tenant.
Project networks (os-tenant-networks)		
POST	/v2.1/{tenant_id}/os-tenant-networks	Creates a project network.
GET	/v2.1/{tenant_id}/os-tenant-networks	Lists all project networks.
GET	/v2.1/{tenant_id}/os-tenant-networks/{network_id}	Shows details for a project network.
DELETE	/v2.1/{tenant_id}/os-tenant-networks/{network_id}	Deletes a project network.
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Associates a network with a host.
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Disassociates a host from a network.
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Disassociates a network from a project so that the network can be reused.
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Disassociates a project from a network.
Volume extension (os-volumes, os-snapshots)		
GET	/v2.1/{tenant_id}/os-volumes	Lists the volumes associated with the account.
POST	/v2.1/{tenant_id}/os-volumes	Creates a volume.
GET	/v2.1/{tenant_id}/os-volumes/detail	Lists all volumes with details.
GET	/v2.1/{tenant_id}/os-volumes/{volume_id}	Shows details for a volume.
DELETE	/v2.1/{tenant_id}/os-volumes/{volume_id}	Deletes a volume.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-volume-types	Lists volume types.
GET	/v2.1/{tenant_id}/os-volume-types/{volume_type_id}	Shows details for a volume type.
POST	/v2.1/{tenant_id}/os-snapshots	Creates a snapshot.
GET	/v2.1/{tenant_id}/os-snapshots	Lists snapshots.
GET	/v2.1/{tenant_id}/os-snapshots/detail	Lists all snapshots with details.
GET	/v2.1/{tenant_id}/os-snapshots/{snapshot_id}	Shows details for a snapshot.
DELETE	/v2.1/{tenant_id}/os-snapshots/{snapshot_id}	Deletes a snapshot from the account.

4.1. API versions

Lists information for all API versions.

Method	URI	Description
GET	/	Lists information about all Compute API versions.
GET	/v2.1	Show information about Compute API v2.1.
GET	/v2	Show information about Compute API v2.

4.1.1. List API versions

Method	URI	Description
GET	/	Lists information about all Compute API versions.

Normal response codes: 200300

4.1.1.1. Request

This operation does not accept a request body.

4.1.1.2. Response

Example 4.1. List API versions: JSON response

```
{
  "versions": [
    {
      "id": "v2.0",
      "links": [
        {
          "href": "http://openstack.example.com/v2/",
          "rel": "self"
        }
      ],
      "status": "SUPPORTED",
      "version": "",
      "min_version": "",
      "updated": "2011-01-21T11:33:21Z"
    },
    {
      "id": "v2.1",
      "links": [
        {
          "href": "http://openstack.example.com/v2.1/",
          "rel": "self"
        }
      ],
      "status": "CURRENT",
      "version": "2.14",
      "min_version": "2.1",
      "updated": "2013-07-23T11:33:21Z"
    }
  ]
}
```

4.1.2. Show API v2.1

Method	URI	Description
GET	/v2.1	Show information about Compute API v2.1.

Normal response codes: 200

4.1.2.1. Request

This operation does not accept a request body.

4.1.2.2. Response

Example 4.2. Show API v2.1: JSON response

```
{
  "version": {
    "id": "v2.1",
    "links": [
      {
        "href": "http://openstack.example.com/v2.1/",
        "rel": "self"
      },
      {
        "href": "http://docs.openstack.org/",
        "rel": "describedby",
        "type": "text/html"
      }
    ],
    "media-types": [
      {
        "base": "application/json",
        "type": "application/vnd.openstack.compute+json;version=2.1"
      }
    ],
    "status": "CURRENT",
    "version": "2.21",
    "min_version": "2.1",
    "updated": "2013-07-23T11:33:21Z"
  }
}
```

4.1.3. Show API v2

Method	URI	Description
GET	/v2	Show information about Compute API v2.

Normal response codes: 200

4.1.3.1. Request

This operation does not accept a request body.

4.1.3.2. Response

Example 4.3. Show API v2: JSON response

```
{
  "version": {
    "id": "v2.0",
    "links": [
      {
        "href": "http://openstack.example.com/v2/",
        "rel": "self"
      },
      {
        "href": "http://docs.openstack.org/",
        "rel": "describedby",
        "type": "text/html"
      }
    ]
  },
  "media-types": [
    {
      "base": "application/json",
      "type": "application/vnd.openstack.compute+json;version=2"
    }
  ],
  "status": "SUPPORTED",
  "version": "",
  "min_version": "",
  "updated": "2011-01-21T11:33:21Z"
}
```

4.2. Servers (servers)

Lists, creates, shows details for, updates, and deletes servers.

Passwords

When you create a server, you can specify a password through the optional `adminPass` attribute. The password must meet the complexity requirements set by your OpenStack Compute provider. The server might enter an ERROR state if the complexity requirements are not met. In this case, a client might issue a change password action to reset the server password.

If you do not specify a password, the API generates and assigns a random password that it returns in the response object. This password meets the security requirements set by the compute provider. For security reasons, subsequent **GET** calls do not require this password.

Server metadata

You can specify custom server metadata at server launch time. The maximum size for each metadata key-value pair is 255 bytes. The compute provider determines the maximum number of key-value pairs for each server. You can query this value through the `maxServerMetadata` absolute limit.

Server networks

You can specify one or more networks to which the server connects at launch time. Users can also specify a specific port on the network or the fixed IP address to assign to the server interface.



Note

You can use both IPv4 and IPv6 addresses as access addresses and you can assign both addresses simultaneously. You can update access addresses after you create a server.

Server personality

To customize the personality of a server instance, you can inject data into its file system. For example, you might insert ssh keys, set configuration files, or store data that you want to retrieve from inside the instance. This customization method provides minimal launch-time personalization. If you require significant customization, create a custom image.

Follow these guidelines when you inject files:

- The maximum size of the file path data is 255 bytes.
- Encode the file contents as a Base64 string. The compute provider determines the maximum size of the file contents. The image that you use to create the server determines this value.



Note

The maximum limit refers to the number of bytes in the decoded data and not to the number of characters in the encoded data.

- The `maxPersonality` absolute limit defines the maximum number of file path and content pairs that you can supply. The compute provider determines this value.
- The `maxPersonalitySize` absolute limit is a byte limit that applies to all images in the deployment. Providers can set additional per-image personality limits.

The file injection might not occur until after the server builds and boots.

After file injection, only system administrators can access personality files. For example, on Linux, all files have root as the owner and the root group as the group owner, and allow only user and group read access (`chmod 440`).

Server access addresses

In a hybrid environment, the underlying implementation might not control the IP address of a server. Instead, the access IP address might be part of the dedicated hardware; for example, a router/NAT device. In this case, you cannot use the addresses that the implementation provides to access the server from outside the local LAN. Instead, the API might assign a separate *access address* at creation time to provide access to the server. This address might not be directly bound to a network interface on the server and might not necessarily appear when you query the server addresses. However, clients should use an access address to access the server directly.

Method	URI	Description
GET	/v2.1/{tenant_id}/servers{? changes-since,image,flavor,name, status,host,limit,marker}	Lists IDs, names, and links for all servers.
POST	/v2.1/{tenant_id}/servers	Creates a server.
GET	/v2.1/{tenant_id}/servers/detail {?changes-since,image,flavor,name, status,host,limit,marker}	Lists all servers with details.
GET	/v2.1/{tenant_id}/servers/ {server_id}	Shows details for a server.
PUT	/v2.1/{tenant_id}/servers/ {server_id}	Updates the editable attributes of a server.
DELETE	/v2.1/{tenant_id}/servers/ {server_id}	Deletes a server.

4.2.1. List servers

Method	URI	Description
GET	/v2.1/{tenant_id}/servers{?changes-since,image,flavor,name,status,host,limit,marker}	Lists IDs, names, and links for all servers.

Servers contain a status attribute that indicates the current server state. You can filter on the server status when you complete a list servers request. The server status is returned in the response body. The possible server status values are:

Server status values

- **ACTIVE.** The server is active.
- **BUILDING.** The server has not finished the original build process.
- **DELETED.** The server is permanently deleted.
- **ERROR.** The server is in error.
- **HARD_REBOOT.** The server is hard rebooting. This is equivalent to pulling the power plug on a physical server, plugging it back in, and rebooting it.
- **MIGRATING.** The server is being migrated to a new host.
- **PASSWORD.** The password is being reset on the server.
- **PAUSED.** In a paused state, the state of the server is stored in RAM. A paused server continues to run in frozen state.
- **REBOOT.** The server is in a soft reboot state. A reboot command was passed to the operating system.
- **REBUILD.** The server is currently being rebuilt from an image.
- **RESCUED.** The server is in rescue mode. A rescue image is running with the original server image attached.
- **RESIZED.** Server is performing the differential copy of data that changed during its initial copy. Server is down for this stage.
- **REVERT_RESIZE.** The resize or migration of a server failed for some reason. The destination server is being cleaned up and the original source server is restarting.
- **SOFT_DELETED.** The server is marked as deleted but the disk images are still available to restore.
- **STOPPED.** The server is powered off and the disk image still persists.
- **SUSPENDED.** The server is suspended, either by request or necessity. This status appears for only the XenServer/XCP, KVM, and ESXi hypervisors. Administrative users can suspend an instance if it is infrequently used or to perform system maintenance. When you

suspend an instance, its VM state is stored on disk, all memory is written to disk, and the virtual machine is stopped. Suspending an instance is similar to placing a device in hibernation; memory and vCPUs become available to create other instances.

- UNKNOWN. The state of the server is unknown. Contact your cloud provider.
- VERIFY_RESIZE. System is awaiting confirmation that the server is operational after a move or resize.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405)

4.2.1.1. Request

This table shows the URI parameters for the list servers request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list servers request:

Name	Type	Description
changes-since	DateTime (Optional)	Filters the response by a date and time stamp when the server last changed status.
image	UUID (Optional)	Filters the response by an image, as a UUID.
flavor	UUID (Optional)	Filters the response by a flavor, as a UUID. A flavor is a combination of memory, disk size, and CPUs.
name	Regex (Optional)	Filters the response by a server name, as a string. You can use regular expressions in the query. For example, the ? name=bob regular expression returns both bob and bobb. If you must match on only bob, you can use a regular expression that matches the syntax of the underlying database server that is implemented for Compute, such as MySQL or PostgreSQL.
status	ServerStatus (Optional)	Filters the response by a server status, as a string. For example, ACTIVE.
host	String (Optional)	Filters the response by a host name, as a string. This query parameter is typically available to only administrative users. If you are a non-administrative user, the API ignores this parameter.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

4.2.1.2. Response

Example 4.4. List servers: JSON response

```
{
  "servers": [
    {
      "id": "a291599e-6de2-41a6-88df-c443ddcef70d",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/
servers/a291599e-6de2-41a6-88df-c443ddcef70d",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/servers/
a291599e-6de2-41a6-88df-c443ddcef70d",
          "rel": "bookmark"
        }
      ],
      "name": "new-server-test"
    }
  ]
}
```

4.2.2. Create server

Method	URI	Description
POST	/v2.1/{tenant_id}/servers	Creates a server.

The progress of this operation depends on the location of the requested image, network I/O, host load, selected flavor, and other factors.

To check the progress of the request, make a GET `/servers/{id}` request. This call returns a progress attribute, which is a percentage value from 0 to 100.

The Location header returns the full URL to the newly created server and is available as a self and bookmark link in the server representation.



Note

When you create a server, the response shows only the server ID, its links, and the admin password. You can get additional attributes through subsequent GET requests on the server.

Include the `block-device-mapping-v2` parameter in the create request body to boot a server from a volume.

Include the `key_name` parameter in the create request body to add a keypair to the server when you create it. To create a keypair, make a [create keypair](#) request.

Preconditions

- The user must have sufficient server quota to create the number of servers requested.
- The connection to the Image service is valid.

Asynchronous postconditions

- With correct permissions, you can see the server status as ACTIVE through API calls.
- With correct access, you can see the created server in the compute node that OpenStack Compute manages.

Troubleshooting

- If the server status remains BUILDING or shows another error status, the request failed. Ensure you meet the preconditions then investigate the compute node.
- The server is not created in the compute node that OpenStack Compute manages.
- The compute node needs enough free resource to match the resource of the server creation request.
- Ensure that the scheduler selection filter can fulfill the request with the available compute nodes that match the selection criteria of the filter.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.2.2.1. Request

This table shows the URI parameters for the create server request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This list shows the body parameters for the request:

- **parameters:**
 - **block_device_mapping_v2:**
 - **device_name:** Xsd:string. Required.
A path to the device for the volume that you want to use to boot the server.
 - **source_type:** Xsd:string. Required.
The source type of the volume. A valid value is blank, snapshot, volume, or image.
 - **destination_type:** Xsd:string. Optional.
Defines where the volume comes from. A valid value is local or volume.
 - **delete_on_termination:** Xsd:string. Required.
To delete the boot volume when the server is destroyed, specify true. Otherwise, specify false.
 - **guest_format:** Xsd:string. Required.
Specifies the guest server disk file system format, such as ephemeral or swap.
 - **boot_index:** Xsd:string. Required.
Defines the order in which a hypervisor tries devices when it attempts to boot the guest from storage.

Give each device a unique boot index starting from 0. To disable a device from booting, set the boot index to a negative value or use the default boot index value, which is None.

The simplest usage is, set the boot index of the boot device to 0 and use the default boot index value, None, for any other devices.

Some hypervisors might not support booting from multiple devices; these hypervisors consider only the device with a boot index of 0.

Some hypervisors support booting from multiple devices but only if the devices are of different types. For example, a disk and CD-ROM.

Example 4.5. Create server: JSON request

```
{
  "server": {
    "name": "new-server-test",
    "imageRef": "http://glance.openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
    "flavorRef": "http://openstack.example.com/flavors/1",
    "metadata": {
      "My Server Name": "Apache1"
    }
  }
}
```

4.2.2.2. Response

Example 4.6. Create server: JSON response

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "adminPass": "zPnp2GseTqG4",
    "id": "8195065c-fea4-4d57-b93f-5c5c63fe90e8",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/8195065c-fea4-4d57-b93f-5c5c63fe90e8",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/8195065c-fea4-4d57-b93f-5c5c63fe90e8",
        "rel": "bookmark"
      }
    ],
    "security_groups": [
      {
        "name": "default"
      }
    ]
  }
}
```

4.2.3. List details for servers

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/detail {?changes-since,image,flavor,name, status,host,limit,marker}	Lists all servers with details.

The compute provisioning algorithm has an anti-affinity property that attempts to spread customer VMs across hosts. Under certain situations, VMs from the same customer might be placed on the same host. The `hostId` property shows the host that your server runs on and can be used to determine this scenario if it is relevant to your application.

For each server, shows server details including configuration drive, extended status, and server usage information.

The extended status information appears in the `OS-EXT-STS:vm_state`, `OS-EXT-STS:power_state`, and `OS-EXT-STS:task_state` attributes.

The server usage information appears in the `OS-SRV-USG:launched_at` and `OS-SRV-USG:terminated_at` attributes.

To hide addresses information for instances in a certain state, set the `osapi_hide_server_address_states` configuration option. Set this option to a valid VM state in the `nova.conf` configuration file.



Note

`HostId` is unique *per account* and is not globally unique.

Normal response codes: 200

Error response codes: `computeFault` (400, 500, ...), `serviceUnavailable` (503), `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405)

4.2.3.1. Request

This table shows the URI parameters for the list details for servers request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list details for servers request:

Name	Type	Description
changes-since	DateTime (Optional)	Filters the response by a date and time stamp when the server last changed status.
image	UUID (Optional)	Filters the response by an image, as a UUID.
flavor	UUID (Optional)	Filters the response by a flavor, as a UUID. A flavor is a combination of memory, disk size, and CPUs.

Name	Type	Description
name	Regexp (Optional)	Filters the response by a server name, as a string. You can use regular expressions in the query. For example, the ? name=bob regular expression returns both bob and bobb. If you must match on only bob, you can use a regular expression that matches the syntax of the underlying database server that is implemented for Compute, such as MySQL or PostgreSQL.
status	ServerStatus (Optional)	Filters the response by a server status, as a string. For example, ACTIVE.
host	String (Optional)	Filters the response by a host name, as a string. This query parameter is typically available to only administrative users. If you are a non-administrative user, the API ignores this parameter.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

4.2.3.2. Response

Example 4.7. List details for servers: JSON response

```
{
  "servers": [
    {
      "addresses": {
        "private": [
          {
            "addr": "192.168.0.3",
            "OS-EXT-IPS-MAC:mac_addr": "aa:bb:cc:dd:ee:ff",
            "OS-EXT-IPS:type": "fixed",
            "version": 4
          }
        ]
      },
      "created": "2013-09-23T13:53:12Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "hostId":
"f1e160ad2bf07084f3d3e0dfdd0795d80da18a60825322c15775c0dd",
      "id": "9cbeffc35-d372-40c5-88e2-9fda1b6ea12c",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
```

```

        {
            "href": "http://openstack.example.com/openstack/
images/70a599e0-31e7-49b7-b260-868f441e862b",
            "rel": "bookmark"
        }
    ],
    "key_name": null,
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/
servers/9cbefc35-d372-40c5-88e2-9fda1b6ea12c",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/servers/
9cbefc35-d372-40c5-88e2-9fda1b6ea12c",
            "rel": "bookmark"
        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "accessIPv4": "",
    "accessIPv6": "",
    "config_drive": "",
    "OS-DCF:diskConfig": "AUTO",
    "OS-EXT-AZ:availability_zone": "nova",
    "OS-EXT-SRV-ATTR:host": "c3f14e9812ad496baf92ccfb3c61e15f",
    "OS-EXT-SRV-ATTR:hypervisor_hostname": "fake-mini",
    "OS-EXT-SRV-ATTR:instance_name": "instance-00000001",
    "OS-EXT-STS:power_state": 1,
    "OS-EXT-STS:task_state": null,
    "OS-EXT-STS:vm_state": "active",
    "os-extended-volumes:volumes_attached": [],
    "OS-SRV-USG:launched_at": "2013-09-23T13:53:12.774549",
    "OS-SRV-USG:terminated_at": null,
    "progress": 0,
    "security_groups": [
        {
            "name": "default"
        }
    ],
    "status": "ACTIVE",
    "host_status": "UP",
    "tenant_id": "openstack",
    "updated": "2013-10-31T06:32:32Z",
    "user_id": "fake"
}
]
}

```

4.2.4. Show server details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}	Shows details for a server.

Includes server details including configuration drive, extended status, and server usage information.

The extended status information appears in the OS-EXT-STS:vm_state, OS-EXT-STS:power_state, and OS-EXT-STS:task_state attributes.

The server usage information appears in the OS-SRV-USG:launched_at and OS-SRV-USG:terminated_at attributes.

To hide addresses information for instances in a certain state, set the `osapi_hide_server_address_states` configuration option. Set this option to a valid VM state in the `nova.conf` configuration file.



Note

HostId is unique *per account* and is not globally unique.

Preconditions

- The server must exist.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.2.4.1. Request

This table shows the URI parameters for the show server details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.2.4.2. Response

Example 4.8. Show server details: JSON response

```
{
  "server": {
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "OS-EXT-IPS-MAC:mac_addr": "aa:bb:cc:dd:ee:ff",
          "OS-EXT-IPS:type": "fixed",
          "version": 4
        }
      ]
    }
  }
}
```



```

    }
  ]
},
"created": "2013-09-23T13:37:00Z",
"flavor": {
  "id": "1",
  "links": [
    {
      "href": "http://openstack.example.com/openstack/flavors/
1",
      "rel": "bookmark"
    }
  ]
},
"hostId": "9cc36101a27c2a69c1a18241f6228454d9d7f466bd90c62db8e8b856",
"id": "f474386b-4fb6-4e1f-b1d5-d6bf4437f7d5",
"image": {
  "id": "70a599e0-31e7-49b7-b260-868f441e862b",
  "links": [
    {
      "href": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
      "rel": "bookmark"
    }
  ]
},
"key_name": null,
"links": [
  {
    "href": "http://openstack.example.com/v2/openstack/servers/
f474386b-4fb6-4e1f-b1d5-d6bf4437f7d5",
    "rel": "self"
  },
  {
    "href": "http://openstack.example.com/openstack/servers/
f474386b-4fb6-4e1f-b1d5-d6bf4437f7d5",
    "rel": "bookmark"
  }
],
"metadata": {
  "My Server Name": "Apache1"
},
"name": "new-server-test",
"accessIPv4": "192.0.2.0",
"accessIPv6": "2002:0:0:0:0:0:c000:20e",
"config_drive": "",
"OS-DCF:diskConfig": "AUTO",
"OS-EXT-AZ:availability_zone": "nova",
"OS-EXT-SRV-ATTR:host": "b8b357f7100d4391828f2177c922ef93",
"OS-EXT-SRV-ATTR:hypervisor_hostname": "fake-mini",
"OS-EXT-SRV-ATTR:instance_name": "instance-00000001",
"OS-EXT-STS:power_state": 1,
"OS-EXT-STS:task_state": null,
"OS-EXT-STS:vm_state": "active",
"os-extended-volumes:volumes_attached": [],
"OS-SRV-USG:launched_at": "2013-09-23T13:37:00.880302",
"OS-SRV-USG:terminated_at": null,
"progress": 0,
"security_groups": [
  {

```

```
        "name": "default"
      }
    ],
    "status": "ACTIVE",
    "host_status": "UP",
    "tenant_id": "openstack",
    "updated": "2013-10-31T07:31:30Z",
    "user_id": "fake"
  }
}
```

4.2.5. Update server

Method	URI	Description
PUT	/v2.1/{tenant_id}/servers/{server_id}	Updates the editable attributes of a server.

Preconditions

- The server must exist.
- You can edit the accessIPv4, accessIPv6, diskConfig and name attributes.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), buildInProgress (409)

4.2.5.1. Request

This table shows the URI parameters for the update server request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.9. Update server name: JSON request

```
{
  "server": {
    "name": "new-server-test",
    "imageRef": "http://glance.openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
    "flavorRef": "http://openstack.example.com/flavors/1",
    "metadata": {
      "My Server Name": "Apache1"
    }
  }
}
```

Example 4.10. Update server IP addresses: JSON request

```
{
  "server": {
    "accessIPv4": "192.0.2.0",
    "accessIPv6": "2002:0:0:0:0:0:c000:20e"
  }
}
```

This list shows the body parameters for the request:

- **parameters:**
 - **block_device_mapping_v2:**

- **device_name**: Xsd:string. Required.

A path to the device for the volume that you want to use to boot the server.

- **source_type**: Xsd:string. Required.

The source type of the volume. A valid value is `blank`, `snapshot`, `volume`, or `image`.

- **destination_type**: Xsd:string. Optional.

Defines where the volume comes from. A valid value is `local` or `volume`.

- **delete_on_termination**: Xsd:string. Required.

To delete the boot volume when the server is destroyed, specify `true`. Otherwise, specify `false`.

- **guest_format**: Xsd:string. Required.

Specifies the guest server disk file system format, such as `ephemeral` or `swap`.

- **boot_index**: Xsd:string. Required.

Defines the order in which a hypervisor tries devices when it attempts to boot the guest from storage.

Give each device a unique boot index starting from 0. To disable a device from booting, set the boot index to a negative value or use the default boot index value, which is `None`.

The simplest usage is, set the boot index of the boot device to 0 and use the default boot index value, `None`, for any other devices.

Some hypervisors might not support booting from multiple devices; these hypervisors consider only the device with a boot index of 0.

Some hypervisors support booting from multiple devices but only if the devices are of different types. For example, a disk and CD-ROM.

Example 4.11. Update server OS-DCF:diskConfig parameter: JSON request

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO"
  }
}
```

4.2.5.2. Response

Example 4.12. Update server name: JSON response

```
{
  "server": {
    "id": "52415800-8b69-11e0-9b19-734f565bc83b",
  }
}
```

```

    "tenant_id": "1234",
    "user_id": "5678",
    "name": "new-server-test",
    "created": "2010-11-11T12:00:00Z",
    "updated": "2010-11-12T12:44:44Z",
    "hostId": "e4d909c290d0fb1ca068ffaddf22cbd0",
    "accessIPv4": "192.0.2.0",
    "accessIPv6": "2002:0:0:0:0:0:c000:20e",
    "progress": 0,
    "status": "ACTIVE",
    "image": {
      "id": "52415800-8b69-11e0-9b19-734f6f006e54",
      "name": "CentOS 5.2",
      "links": [
        {
          "rel": "self",
          "href": "http://servers.api.openstack.org/v2/1234/images/52415800-8b69-11e0-9b19-734f6f006e54"
        },
        {
          "rel": "bookmark",
          "href": "http://servers.api.openstack.org/1234/images/52415800-8b69-11e0-9b19-734f6f006e54"
        }
      ]
    },
    "flavor": {
      "id": "52415800-8b69-11e0-9b19-734f1195ff37",
      "name": "256 MB Server",
      "links": [
        {
          "rel": "self",
          "href": "http://servers.api.openstack.org/v2/1234/flavors/52415800-8b69-11e0-9b19-734f1195ff37"
        },
        {
          "rel": "bookmark",
          "href": "http://servers.api.openstack.org/1234/flavors/52415800-8b69-11e0-9b19-734f1195ff37"
        }
      ]
    },
    "metadata": {
      "My Server Name": "Apache1"
    },
    "addresses": {
      "public": [
        {
          "version": 4,
          "addr": "192.0.2.0"
        },
        {
          "version": 6,
          "addr": "2002:0:0:0:0:0:c000:20e"
        }
      ],
      "private": [
        {
          "version": 4,
          "addr": "198.51.100.0"
        }
      ]
    }
  }
}

```

```
        },
        {
            "version": 6,
            "addr": "2002:0:0:0:0:c633:640e"
        }
    ],
    },
    "links": [
        {
            "rel": "self",
            "href": "http://servers.api.openstack.org/v2/1234/servers/52415800-8b69-11e0-9b19-734fcede0043"
        },
        {
            "rel": "bookmark",
            "href": "http://servers.api.openstack.org/1234/servers/52415800-8b69-11e0-9b19-734fcede0043"
        }
    ]
    }
}
```

4.2.6. Delete server

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}	Deletes a server.

Preconditions

- The server must exist.
- Anyone can delete a server when the status of the server is not locked and when the policy allows.
- If the server is locked, you must have administrator privileges to delete the server.

Asynchronous postconditions

- With correct permissions, you can see the server status as DELETED through API calls.
- The port attached to the server is deleted.
- The server does not appear in the list servers response.
- The server managed by OpenStack Compute is deleted on the compute node.

Troubleshooting

- If server status remains in deleting status or another error status, the request failed. Ensure that you meet the preconditions. Then, investigate the compute back end.
- The request returns the HTTP 409 response code when the server is locked even if you have correct permissions. Ensure that you meet the preconditions then investigate the server status.
- The server managed by OpenStack Compute is not deleted from the compute node.

Normal response codes: 204

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.2.6.1. Request

This table shows the URI parameters for the delete server request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.3. Servers multiple create (servers)

Creates one or more servers.

Optionally, you can set "return_reservation_id": "True" in the request body to request that a reservation ID be returned instead of the newly created instance information. With this parameter, the response shows only the reservation ID.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers	Creates one or more servers.
POST	/v2.1/{tenant_id}/servers	Creates one or more servers with a reservation ID.

4.3.1. Create multiple servers

Method	URI	Description
POST	/v2.1/{tenant_id}/servers	Creates one or more servers.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.3.1.1. Request

This table shows the URI parameters for the create multiple servers request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This list shows the body parameters for the request:

- **parameters:**
 - **block_device_mapping_v2:**
 - **device_name:** Xsd:string. Required.
A path to the device for the volume that you want to use to boot the server.
 - **source_type:** Xsd:string. Required.
The source type of the volume. A valid value is `blank`, `snapshot`, `volume`, or `image`.
 - **destination_type:** Xsd:string. Optional.
Defines where the volume comes from. A valid value is `local` or `volume`.
 - **delete_on_termination:** Xsd:string. Required.
To delete the boot volume when the server is destroyed, specify `true`. Otherwise, specify `false`.
 - **guest_format:** Xsd:string. Required.
Specifies the guest server disk file system format, such as `ephemeral` or `swap`.
 - **boot_index:** Xsd:string. Required.
Defines the order in which a hypervisor tries devices when it attempts to boot the guest from storage.

Give each device a unique boot index starting from 0. To disable a device from booting, set the boot index to a negative value or use the default boot index value, which is None.

The simplest usage is, set the boot index of the boot device to 0 and use the default boot index value, None, for any other devices.

Some hypervisors might not support booting from multiple devices; these hypervisors consider only the device with a boot index of 0.

Some hypervisors support booting from multiple devices but only if the devices are of different types. For example, a disk and CD-ROM.

Example 4.13. Create multiple servers without reservation ID

```
{
  "server": {
    "name": "new-server-test",
    "imageRef": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
    "flavorRef": "http://openstack.example.com/openstack/flavors/1",
    "metadata": {
      "My Server Name": "Apache1"
    },
    "min_count": "2",
    "max_count": "3"
  }
}
```

4.3.1.2. Response

Example 4.14. Create multiple servers without reservation ID

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "adminPass": "zPnp2GseTqG4",
    "id": "8195065c-fea4-4d57-b93f-5c5c63fe90e8",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/8195065c-fea4-4d57-b93f-5c5c63fe90e8",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/8195065c-fea4-4d57-b93f-5c5c63fe90e8",
        "rel": "bookmark"
      }
    ],
    "security_groups": [
      {
        "name": "default"
      }
    ]
  }
}
```


4.3.2. Create multiple servers with reservation ID

Method	URI	Description
POST	/v2.1/{tenant_id}/servers	Creates one or more servers with a reservation ID.

Set "return_reservation_id": "True" in the request body to request that a reservation ID be returned instead of the newly created instance information. With this parameter, the response shows only the reservation ID.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.3.2.1. Request

This table shows the URI parameters for the create multiple servers with reservation id request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This list shows the body parameters for the request:

- **parameters:**
 - **block_device_mapping_v2:**
 - **device_name:** Xsd:string. Required.
A path to the device for the volume that you want to use to boot the server.
 - **source_type:** Xsd:string. Required.
The source type of the volume. A valid value is blank, snapshot, volume, or image.
 - **destination_type:** Xsd:string. Optional.
Defines where the volume comes from. A valid value is local or volume.
 - **delete_on_termination:** Xsd:string. Required.
To delete the boot volume when the server is destroyed, specify true. Otherwise, specify false.
 - **guest_format:** Xsd:string. Required.
Specifies the guest server disk file system format, such as ephemeral or swap.
 - **boot_index:** Xsd:string. Required.

Defines the order in which a hypervisor tries devices when it attempts to boot the guest from storage.

Give each device a unique boot index starting from 0. To disable a device from booting, set the boot index to a negative value or use the default boot index value, which is None.

The simplest usage is, set the boot index of the boot device to 0 and use the default boot index value, None, for any other devices.

Some hypervisors might not support booting from multiple devices; these hypervisors consider only the device with a boot index of 0.

Some hypervisors support booting from multiple devices but only if the devices are of different types. For example, a disk and CD-ROM.

Example 4.15. Create multiple servers with reservation ID

```
{
  "server": {
    "name": "new-server-test",
    "imageRef": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
    "flavorRef": "http://openstack.example.com/openstack/flavors/1",
    "metadata": {
      "My Server Name": "Apache1"
    },
    "return_reservation_id": true,
    "min_count": "2",
    "max_count": "3"
  }
}
```

4.3.2.2. Response

Example 4.16. Create multiple servers with reservation ID

```
{
  "reservation_id": "r-3fhpjulah"
}
```

4.4. Servers - run an action (servers, action)

Enables all users to perform an action on a server. Specify the action in the request body.

You can associate a fixed or floating IP address with a server instance, or disassociate a fixed or floating IP address from a server instance. You can attach a volume to a server.

You can create an image from a server, evacuate a server from a failed host to a new host, and force-delete a server before deferred cleanup. You can lock, pause, reboot, rebuild, rescue, resize, resume, confirm the resize of, revert a pending resize for, shelve, shelf-offload, unshelve, start, stop, unlock, unpause, and unrescue a server. You can also change the password of the server and add a security group to or remove a security group from a server. You can also trigger a crash dump into a server since Mitaka release.

You can get an RDP, serial, SPICE, or VNC console for a server.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a fixed IP address to a server instance, which associates that address with the server. The fixed IP address is retrieved from the network that you specify in the request.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to a server, which associates that address with the server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Changes the administrative password for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Confirms a pending resize action for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Creates an image from a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Evacuates a server from a failed host to a new one.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Force-deletes a server before deferred cleanup.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Locks a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Pauses a server. Changes its status to PAUSED.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Shows console output for a server instance.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets an RDP console for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a serial console for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a SPICE console for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a VNC console for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Reboots a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Rebuilds a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Removes, or disassociates, a fixed IP address from a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Removes, or disassociates, a floating IP address from a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Puts a server in rescue mode and changes its status to RESCUE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resizes a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Restores a previously soft-deleted server instance. You cannot use this method to restore deleted instances.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resumes a suspended server and changes its status to ACTIVE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Cancels and reverts a pending resize action for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Shelves a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Shelf-offloads, or removes, a shelved server.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Starts a stopped server and changes its status to ACTIVE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Stops a running server and changes its status to SHUTOFF.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unlocks a locked server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unpauses a paused server and changes its status to ACTIVE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unrescues a server. Changes status to ACTIVE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unshelves, or restores, a shelved server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a security group to a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Removes a security group from a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Trigger a crash dump in a server.

4.4.1. Add (associate) fixed IP (addFixedIp action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a fixed IP address to a server instance, which associates that address with the server. The fixed IP address is retrieved from the network that you specify in the request.

Specify the addFixedIp action and the network ID in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), serverCapacityUnavailable (503), buildInProgress (409)

4.4.1.1. Request

This table shows the URI parameters for the add (associate) fixed ip (addfixedip action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.17. Add (associate) fixed IP (addFixedIp action): JSON request

```
{
  "addFixedIp": {
    "networkId": 1
  }
}
```


4.4.2. Add (associate) floating IP (addFloatingIp action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to a server, which associates that address with the server.

A pool of floating IP addresses, configured by the cloud administrator, is available in OpenStack Compute. The project quota defines the maximum number of floating IP addresses that you can allocate to the project. After you [create \(allocate\) a floating IP address](#) for a project, you can associate that address with the server. Specify the `addFloatingIp` action in the request body.

If an instance is connected to multiple networks, you can associate a floating IP address with a specific fixed IP address by using the optional `fixed_address` parameter.

Preconditions

- The server must exist.
- You can only add a floating IP address to the server when its status is `available`.

Normal response codes: 202

Error response codes: `computeFault` (400, 500, ...), 409, `serviceUnavailable` (503), `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `itemNotFound` (404), `badMediaType` (415), `NetworkNotFound` (400), `serverCapacityUnavailable` (503), `buildInProgress` (409)

4.4.2.1. Request

This table shows the URI parameters for the add (associate) floating ip (`addfloatingip` action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.18. Add (associate) floating IP (addFloatingIp action): JSON request

```
{
  "addFloatingIp": {
    "address": "172.24.4.4"
  }
}
```

4.4.3. Change administrative password (changePassword action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Changes the administrative password for a server.

Specify the changePassword action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the policy.json file.

Normal response codes: 202

4.4.3.1. Request

This table shows the URI parameters for the change administrative password (changePassword action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.19. Change administrative password (changePassword action): JSON request

```
{
  "changePassword": {
    "adminPass": "foo"
  }
}
```

4.4.4. Confirm resized server (confirmResize action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Confirms a pending resize action for a server.

Specify the `confirmResize` action in the request body.

After you make this request, you typically must keep polling the server status to determine whether the request succeeded. A successfully confirming resize operation shows a status of `ACTIVE` or `SHUTOFF` and a `migration_status` of `confirmed`. You can also see the resized server in the compute node that OpenStack Compute manages.

Preconditions

- You can only confirm the resized server where the status is `VERIFY_RESIZED`, the `vm_status` is `RESIZED`, and the `migration_status` is `finished` or `confirming`.
- If the server is locked, you must have administrator privileges to confirm the server.

Troubleshooting

- If the server status remains `RESIZED`, the request failed. Ensure you meet the preconditions and run the request again. If the request fails again, investigate the compute back end or ask your cloud provider.

Normal response codes: 204

Error response codes: `computeFault` (400, 500, ...), `serviceUnavailable` (503), `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `itemNotFound` (404), `badMediaType` (415), `NetworkNotFound` (400), `serverCapacityUnavailable` (503), `buildInProgress` (409)

4.4.4.1. Request

This table shows the URI parameters for the confirm resized server (`confirmresize` action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.20. Confirm resized server (confirmResize action): JSON request

```
{
  "confirmResize": null
}
```

4.4.5. Create image (createImage action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Creates an image from a server.

Specify the createImage action in the request body.

After you make this request, you typically must keep polling the status of the created image to determine whether the request succeeded.

If the operation succeeds, the created image has a status of `active` and the server status returns to the original status. You can also see the new image in the image back end that OpenStack Image service manages.

Preconditions

- The server must exist.
- You can only create a new image from the server when its status is `ACTIVE`, `SHUTOFF`, `PAUSED`, or `SUSPENDED`.
- The connection to the Image service is valid.

Troubleshooting

- If the image status remains uploading or shows another error status, the request failed. Ensure you meet the preconditions and run the request again. If the request fails again, investigate the image back end.
- If the server status does not go back to an original server's status, the request failed. Ensure you meet the preconditions, or check if there is another operation that causes race conditions for the server, then run the request again. If the request fails again, investigate the compute back end or ask your cloud provider.
- If the request fails due to an error on OpenStack Compute service, the image is purged from the image store that OpenStack Image service manages. Ensure you meet the preconditions and run the request again. If the request fails again, investigate OpenStack Compute service or ask your cloud provider.

Normal response codes: 202

Error response codes: `computeFault` (400, 500, ...), `409`, `serviceUnavailable` (503), `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `itemNotFound` (404), `badMediaType` (415), `NetworkNotFound` (400), `serverCapacityUnavailable` (503), `buildInProgress` (409)

4.4.5.1. Request

This table shows the URI parameters for the create image (createimage action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Name	Type	Description
{server_id}	UUID	The UUID of the server.

Example 4.21. Create image (createImage action): JSON request

```
{
  "createImage": {
    "name": "foo-image",
    "metadata": {
      "meta_var": "meta_val"
    }
  }
}
```

4.4.6. Evacuate server (evacuate action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Evacuates a server from a failed host to a new one.

Specify the evacuate action in the request body.

Normal response codes: 202

4.4.6.1. Request

This table shows the URI parameters for the evacuate server (evacuate action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.22. Evacuate server (evacuate action): JSON request

```
{
  "evacuate": {
    "host": "b419863b7d814906a68fb31703c0dbd6",
    "adminPass": "MySecretPass"
  }
}
```

4.4.6.2. Response

Example 4.23. Evacuate server (evacuate action): JSON response

```
{
  "adminPass": "MySecretPass"
}
```

4.4.7. Force-delete server (forceDelete action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Force-deletes a server before deferred cleanup.

Specify the `forceDelete` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.4.7.1. Request

This table shows the URI parameters for the force-delete server (`forcedelete` action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.24. Force-delete server (forceDelete action): JSON request

```
{
  "forceDelete": null
}
```

4.4.8. Lock server (lock action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Locks a server.

Specify the lock action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.4.8.1. Request

This table shows the URI parameters for the lock server (lock action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.25. Lock server (lock action): JSON request

```
{
  "lock": null
}
```


4.4.9. Pause server (pause action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Pauses a server. Changes its status to PAUSED.

Specify the pause action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.4.9.1. Request

This table shows the URI parameters for the pause server (pause action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.26. Pause server (pause action): JSON request

```
{
  "pause": null
}
```

4.4.10. Show console output (os-getConsoleOutput action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Shows console output for a server instance.

Specify the `os-getConsoleOutput` action in the request body.

Normal response codes: 200

4.4.10.1. Request

This table shows the URI parameters for the show console output (`os-getconsoleoutput` action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.27. Get console output: JSON request

```
{
  "os-getConsoleOutput": {
    "length": 50
  }
}
```

4.4.10.2. Response

Example 4.28. Get console output: JSON response

```
{
  "output": "FAKE CONSOLE OUTPUT\nANOTHER\nLAST LINE"
}
```

4.4.11. Get RDP console (os-getRDPConsole action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets an RDP console for a server.

Specify the `os-getRDPConsole` action in the request body.

Normal response codes: 200

4.4.11.1. Request

This table shows the URI parameters for the `get rdp console` (`os-getrdpconsole` action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.29. Get RDP console (os-getRDPConsole action): JSON request

```
{
  "os-getRDPConsole": {
    "type": "rdp-html5"
  }
}
```

4.4.11.2. Response

Example 4.30. Get RDP console (os-getRDPConsole action): JSON response

```
{
  "console": {
    "type": "rdp-html5",
    "url": "http://127.0.0.1:6083/?token=191996c3-7b0f-42f3-95a7-f1839f2da6ed"
  }
}
```

4.4.12. Get serial console (os-getSerialConsole action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a serial console for a server.

Specify the `os-getSerialConsole` action in the request body.

Normal response codes: 200

4.4.12.1. Request

This table shows the URI parameters for the get serial console (`os-getserialconsole` action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.31. Get serial console (os-getSerialConsole action): JSON request

```
{
  "os-getSerialConsole": {
    "type": "serial"
  }
}
```

4.4.12.2. Response

Example 4.32. Get serial console (os-getSerialConsole action): JSON response

```
{
  "console": {
    "type": "serial",
    "url": "ws://127.0.0.1:6083/?token=f9906a48-b71e-4f18-baca-c987da3ebdb3"
  }
}
```

4.4.13. Get SPICE console (os-getSPICEConsole action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a SPICE console for a server.

Specify the os-getSPICEConsole action in the request body.

Normal response codes: 200

4.4.13.1. Request

This table shows the URI parameters for the get spice console (os-getspiceconsole action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.33. Get SPICE console (os-getSPICEConsole action): JSON request

```
{
  "os-getSPICEConsole": {
    "type": "spice-html5"
  }
}
```

4.4.13.2. Response

Example 4.34. Get SPICE console (os-getSPICEConsole action): JSON response

```
{
  "console": {
    "type": "spice-html5",
    "url": "http://127.0.0.1:6082/spice_auto.html?token=a30e5d08-6a20-4043-958f-0852440c6af4"
  }
}
```

4.4.14. Get VNC console (os-getVNCConsole action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a VNC console for a server.

Specify the `os-getVNCConsole` action in the request body.

Normal response codes: 200

4.4.14.1. Request

This table shows the URI parameters for the `get vnc console (os-getvncconsole action)` request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.35. Get VNC console (os-getVNCConsole action): JSON request

```
{
  "os-getVNCConsole": {
    "type": "novnc"
  }
}
```

4.4.14.2. Response

Example 4.36. Get VNC console (os-getVNCConsole action): JSON response

```
{
  "console": {
    "type": "novnc",
    "url": "http://127.0.0.1:6080/vnc_auto.html?token=191996c3-7b0f-42f3-95a7-f1839f2da6ed"
  }
}
```

4.4.15. Reboot server (reboot action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Reboots a server.

Specify the reboot action in the request body.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), HTTPUnprocessableEntity (422), buildInProgress (409)

4.4.15.1. Request

This table shows the URI parameters for the reboot server (reboot action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.37. Reboot server (reboot action): JSON request

```
{
  "reboot": {
    "type": "HARD"
  }
}
```

4.4.16. Rebuild server (rebuild action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Rebuilds a server.

Specify the `rebuild` action in the request body.

To rebuild the server with preservation of the ephemeral partition, set the `preserve_ephemeral` parameter to `true`.

Normal response codes: 202

Error response codes: `computeFault` (400, 500, ...), `serviceUnavailable` (503), `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `itemNotFound` (404), `badMediaType` (415), `NetworkNotFound` (400), `serverCapacityUnavailable` (503), `buildInProgress` (409)

4.4.16.1. Request

This table shows the URI parameters for the rebuild server (rebuild action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.38. Rebuild server (rebuild action): JSON request

```
{
  "rebuild": {
    "imageRef": "http://glance.openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
    "name": "foobar",
    "adminPass": "seekr3t",
    "metadata": {
      "meta_var": "meta_val"
    },
    "personality": [
      {
        "path": "/etc/banner.txt",
        "contents": "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5kIGF0IHN1Y2ggYSBzcGVlZC4uLk0IGZlZWxzIGFuIGltcHVsY2lubi4uLnRoXMGaXMGdGhliHBsYWNlIHRvIGdvIG5vdy4gQnV0IHRoZSBza3kga25vd3MgdGhliHJlYXNvbnMgYW5kIHRoZSBwYXR0ZXJucyBiZWpibmQgYWxsIGNsb3VkcycwYmV5b25kIGhvcml6b25zLiINCg0KLvJpY2hhcmQgQmFjaA=="
      }
    ],
    "preserve_ephemeral": true
  }
}
```


}

4.4.16.2. Response

Example 4.39. Rebuild server (rebuild action): JSON response

```
{
  "server": {
    "accessIPv4": "192.0.2.0",
    "accessIPv6": "2002:0:0:0:0:0:c000:20e",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "adminPass": "seekr3t",
    "created": "2013-11-14T06:29:00Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "28d8d56f0e3a77e20891f455721cbb68032e017045e20aa5dfc6cb66",
    "id": "a0a80a94-3d81-4a10-822a-daa0cf9e870b",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    },
    "links": [
      {
        "href": "http://openstack.example.com/v2.1/servers/a0a80a94-3d81-4a10-822a-daa0cf9e870b",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/a0a80a94-3d81-4a10-822a-daa0cf9e870b",
        "rel": "bookmark"
      }
    ],
    "metadata": {
      "meta_var": "meta_val"
    },
    "name": "foobar",
    "progress": 0,
    "status": "ACTIVE",
  }
}
```

```
    "tenant_id": "openstack",  
    "updated": "2013-11-14T06:29:02Z",  
    "user_id": "fake"  
  }  
}
```

4.4.17. Remove (disassociate) fixed IP (removeFixedIp action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Removes, or disassociates, a fixed IP address from a server.

Specify the `removeFixedIp` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

Error response codes: `computeFault` (400, 500, ...), `serviceUnavailable` (503), `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `itemNotFound` (404), `badMediaType` (415), `NetworkNotFound` (400), `serverCapacityUnavailable` (503), `buildInProgress` (409)

4.4.17.1. Request

This table shows the URI parameters for the remove (disassociate) fixed ip (`removefixedip` action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.40. Remove (disassociate) fixed IP (removeFixedIp action): JSON request

```
{
  "removeFixedIp": {
    "address": "10.0.0.4"
  }
}
```

4.4.18. Remove (disassociate) floating IP (removeFloatingIp action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Removes, or disassociates, a floating IP address from a server.

The IP address is returned to the pool of IP addresses that is available for all projects. When you remove a floating IP address and that IP address is still associated with a running instance, it is automatically disassociated from that instance.

Specify the `removeFloatingIp` action in the request body.

Normal response codes: 202

Error response codes: `computeFault` (400, 500, ...), 409, `serviceUnavailable` (503), `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `itemNotFound` (404), `badMediaType` (415), `NetworkNotFound` (400), `serverCapacityUnavailable` (503), `buildInProgress` (409)

4.4.18.1. Request

This table shows the URI parameters for the remove (disassociate) floating ip (removefloatingip action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.41. Remove (disassociate) floating IP (removeFloatingIp action): JSON request

```
{
  "removeFloatingIp": {
    "address": "172.24.4.4"
  }
}
```

4.4.19. Rescue server (rescue action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Puts a server in rescue mode and changes its status to RESCUE.

Specify the rescue action in the request body.

If you specify the `rescue_image_ref` extended attribute, the image is used to rescue the instance. If you omit an image reference, the base image reference is used by default.

Asynchronous Postconditions

- After you successfully rescue a server and make a GET `/v2.1/{tenant_id}/servers/{server_id}` request, its status changes to RESCUE.

Normal response codes: 202

4.4.19.1. Request

This table shows the URI parameters for the rescue server (rescue action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.42. Rescue server: JSON request

```
{
  "rescue": {
    "adminPass": "MySecretPass"
  }
}
```

Example 4.43. Rescue server with image: JSON request

```
{
  "rescue": {
    "adminPass": "MySecretPass",
    "rescue_image_ref": "70a599e0-31e7-49b7-b260-868f441e862b"
  }
}
```

4.4.19.2. Response

Example 4.44. Extended rescue server: JSON response

```
{
  "adminPass": "MySecretPass"
}
```

4.4.20. Resize server (resize action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resizes a server.

Specify the resize action in the request body.

A successfully resized server shows a VERIFY_RESIZE status, RESIZED VM status, and finished migration status. If you set the `resize_confirm_window` option of the Compute service to an integer value, the Compute service automatically confirms the resize operation after the set interval in seconds.

Preconditions

- You can only resize a server when its status is ACTIVE or SHUTOFF.
- If the server is locked, you must have administrator privileges to resize the server.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), serverCapacityUnavailable (503), buildInProgress (409)

4.4.20.1. Request

This table shows the URI parameters for the resize server (resize action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.45. Resize server (resize action): JSON request

```
{
  "resize": {
    "flavorRef": "2"
  }
}
```

4.4.21. Restore soft-deleted instance (restore action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Restores a previously soft-deleted server instance. You cannot use this method to restore deleted instances.

Specify the restore action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.4.21.1. Request

This table shows the URI parameters for the restore soft-deleted instance (restore action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.46. Restore soft-deleted instance (restore action): JSON request

```
{
  "restore": null
}
```

4.4.22. Resume suspended server (resume action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resumes a suspended server and changes its status to ACTIVE.

Specify the resume action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.4.22.1. Request

This table shows the URI parameters for the resume suspended server (resume action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.47. Resume suspended server (resume action): JSON request

```
{
  "resume": null
}
```


4.4.23. Revert resized server (revertResize action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Cancels and reverts a pending resize action for a server.

Specify the `revertResize` action in the request body.

After you make this request, you typically must keep polling the server status to determine whether the request succeeded. A successfully reverting resize operation shows a status of `ACTIVE` or `SHUTOFF` and a `migration_status` of `reverted`. You can also see the reverted server in the compute node that OpenStack Compute manages.

Preconditions

- You can only confirm the resized server where the status is `VERIFY_RESIZE` and the `vm_status` is `RESIZED`.
- If the server is locked, you must have administrator privileges to revert the resizing.

Troubleshooting

- If the server status remains `RESIZED`, the request failed. Ensure you meet the preconditions and run the request again. If the request fails again, investigate the compute back end.
- The server is not reverted in the compute node that OpenStack Compute manages.

Normal response codes: 202

Error response codes: `computeFault` (400, 500, ...), `serviceUnavailable` (503), `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `itemNotFound` (404), `badMediaType` (415), `NetworkNotFound` (400), `serverCapacityUnavailable` (503), `buildInProgress` (409)

4.4.23.1. Request

This table shows the URI parameters for the revert resized server (`revertresize` action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.48. Revert resized server (revertResize action): JSON request

```
{
  "revertResize": null
}
```

4.4.24. Shelf server (shelve action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Shelves a server.

Specify the `shelve` action in the request body.

All associated data and resources are kept but anything still in memory is not retained. To restore a shelved instance, use the `unshelve` action. To remove a shelved instance, use the `shelveOffload` action.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Preconditions

- The server status must be `ACTIVE`, `SHUTOFF`, `PAUSED`, or `SUSPENDED`.
- If the server is locked, you must have administrator privileges to shelve the server.

Asynchronous Postconditions

- After you successfully shelve a server, its status changes to `SHELVED` and the image status is `ACTIVE`. The server instance data appears on the compute node that the Compute service manages.
- If you boot the server from volumes or set the `shelved_offload_time` option to 0, the Compute service automatically deletes the instance on compute nodes and changes the server status to `SHELVED_OFFLOADED`.

Troubleshooting

- If the server status does not change to `SHELVED` or `SHELVED_OFFLOADED`, the shelve operation failed. Ensure that you meet the preconditions and run the request again. If the request fails again, investigate whether another operation is running that causes a race condition.

Normal response codes: 202

4.4.24.1. Request

This table shows the URI parameters for the shelf server (shelve action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.49. Shelf server: JSON request

```
{
```

```
"shelve": null  
}
```

4.4.25. Shelf-offload (remove) server (shelveOffload action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Shelf-offloads, or removes, a shelved server.

Specify the `shelveOffload` action in the request body.

Data and resource associations are deleted. If an instance is no longer needed, you can remove that instance from the hypervisor to minimize resource usage.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Preconditions

- The server status must be `SHELVED`.
- If the server is locked, you must have administrator privileges to shelve-offload the server.

Asynchronous Postconditions

- After you successfully shelve-offload a server, its status changes to `SHELVED_OFFLOADED`. The server instance data appears on the compute node.

Troubleshooting

- If the server status does not change to `SHELVED_OFFLOADED`, the shelve-offload operation failed. Ensure that you meet the preconditions and run the request again. If the request fails again, investigate whether another operation is running that causes a race condition.

Normal response codes: 202

4.4.25.1. Request

This table shows the URI parameters for the shelf-offload (remove) server (`shelveoffload` action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.50. Shelf server: JSON request

```
{
  "shelveOffload": null
}
```

4.4.26. Start server (os-start action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Starts a stopped server and changes its status to ACTIVE.

Specify the `os-start` action in the request body.

Preconditions

- The server status must be SHUTOFF.
- If the server is locked, you must have administrator privileges to start the server.

Asynchronous Postconditions

- After you successfully start a server, its status changes to ACTIVE. The server appears on the compute node that the Compute service manages.

Troubleshooting

- If the server status does not change to ACTIVE, the start operation failed. Ensure that you meet the preconditions and run the request again. If the request fails again, investigate whether another operation is running that causes a race condition.

Normal response codes: 202

4.4.26.1. Request

This table shows the URI parameters for the start server (os-start action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.51. Start server: JSON request

```
{
  "os-start": null
}
```

4.4.27. Stop server (os-stop action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Stops a running server and changes its status to SHUTOFF.

Specify the os-stop action in the request body.

Preconditions

- The server status must be ACTIVE or ERROR.
- If the server is locked, you must have administrator privileges to stop the server.

Asynchronous Postconditions

- After you successfully stop a server, its status changes to SHUTOFF. The server instance data appears only on the compute node that Compute service manages.

Normal response codes: 202

4.4.27.1. Request

This table shows the URI parameters for the stop server (os-stop action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.52. Stop server: JSON request

```
{
  "os-stop": null
}
```

4.4.28. Unlock server (unlock action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unlocks a locked server.

Specify the unlock action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.4.28.1. Request

This table shows the URI parameters for the unlock server (unlock action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.53. Unlock server (unlock action): JSON request

```
{
  "unlock": null
}
```

4.4.29. Unpause server (unpause action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unpauses a paused server and changes its status to ACTIVE.

Specify the unpause action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.4.29.1. Request

This table shows the URI parameters for the unpause server (unpause action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.54. Unpause server (unpause action): JSON request

```
{
  "unpause": null
}
```


4.4.30. Unrescue server (unrescue action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unrescues a server. Changes status to ACTIVE.

Specify the unrescue action in the request body.

Preconditions

- The server must exist.
- You can only unrescue a server when its status is RESCUE.

Asynchronous Postconditions

- After you successfully unrescue a server and make a GET /v2.1/{tenant_id}/servers/{server_id} request, its status changes to ACTIVE.

Normal response codes: 202

4.4.30.1. Request

This table shows the URI parameters for the unrescue server (unrescue action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.55. Unrescue server: JSON request

```
{
  "unrescue": null
}
```

4.4.31. Unshelve (restore) shelved server (unshelve action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unshelves, or restores, a shelved server.

Specify the unshelve action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Preconditions

- The server status must be `SHELVED` or `SHELVED_OFFLOADED`.
- If the server is locked, you must have administrator privileges to unshelve the server.

Asynchronous Postconditions

- After you successfully shelve a server, its status changes to `ACTIVE`. The server appears on the compute node.
- The shelved image is deleted from the list of images returned by an API call.

Troubleshooting

- If the server status does not change to `ACTIVE`, the unshelve operation failed. Ensure that you meet the preconditions and run the request again. If the request fails again, investigate whether another operation is running that causes a race condition.

Normal response codes: 202

4.4.31.1. Request

This table shows the URI parameters for the unshelve (restore) shelved server (unshelve action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.56. Unshelve server: JSON request

```
{
  "unshelve": null
}
```

4.4.32. Add security group to a server (addSecurityGroup action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a security group to a server.

Specify the addSecurityGroup action in the request body.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), badMediaType (415), NetworkNotFound (400)

4.4.32.1. Request

This table shows the URI parameters for the add security group to a server (addsecurity-group action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.57. Add security group to a server (addSecurityGroup action): JSON request

```
{
  "addSecurityGroup": {
    "name": "default"
  }
}
```

4.4.33. Remove security group from a server (removeSecurityGroup action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Removes a security group from a server.

Specify the removeSecurityGroup action in the request body.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), badMediaType (415), NetworkNotFound (400)

4.4.33.1. Request

This table shows the URI parameters for the remove security group from a server (removeSecurityGroup action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.58. Remove security group from a server (removeSecurityGroup action): JSON request

```
{
  "removeSecurityGroup": {
    "name": "default"
  }
}
```

4.4.34. Trigger crash dump in server (trigger_crash_dump action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Trigger a crash dump in a server.

Specify the `trigger_crash_dump` action in the request body.

When a crash dump is triggered for a virtual server, it causes a system reboot. This action can cause data loss. Also, network connectivity can be lost. Once the server comes back on-line, you can find a Kernel Crash Dump file in a certain location of the filesystem. For example, for Ubuntu you can find it in the `/var/crash` directory.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.4.34.1. Request

This table shows the URI parameters for the trigger crash dump in server (trigger_crash_dump action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.59. Trigger crash dump: JSON request

```
{
  "trigger_crash_dump": null
}
```

4.5. Servers - run an administrative action (servers, action)

Enables administrators to perform an action on a server. Specify the action in the request body.

You can change the administrative password for, create a back up of, and inject network information into a server.

You can migrate, live-migrate, reset networking on, and reset the state of a server.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Creates a back up of a server.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Injects network information into a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Migrates a server to a host. The scheduler chooses the host.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Live-migrates a server to a new host without rebooting.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resets networking on a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resets the state of a server.

4.5.1. Create server back up (createBackup action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Creates a back up of a server.

Specify the createBackup action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.5.1.1. Request

This table shows the URI parameters for the create server back up (createbackup action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.60. Create server back up (createBackup action): JSON request

```
{
  "createBackup": {
    "name": "Backup 1",
    "backup_type": "daily",
    "rotation": 1
  }
}
```

4.5.2. Inject network information (injectNetworkInfo action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Injects network information into a server.

Specify the `injectNetworkInfo` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.5.2.1. Request

This table shows the URI parameters for the inject network information (`injectnetworkinfo` action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.61. Inject network information (injectNetworkInfo action): JSON request

```
{
  "injectNetworkInfo": null
}
```


4.5.3. Migrate server (migrate action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Migrates a server to a host. The scheduler chooses the host.

Specify the migrate action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.5.3.1. Request

This table shows the URI parameters for the migrate server (migrate action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.62. Migrate server (migrate action): JSON request

```
{
  "migrate": null
}
```

4.5.4. Live-migrate server (os-migrateLive action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Live-migrates a server to a new host without rebooting.

Use the `host` parameter to specify the destination host. If you omit this parameter, the scheduler chooses a host. If a scheduled host is not suitable, the scheduler tries up to `migrate_max_retries` rescheduling attempts.

If both source and destination hosts provide local disks, you can set the `block_migration` parameter to `true`. If either host uses shared storage, the migration fails if you set this parameter to `true`.

Policy defaults enable only users with the administrative role to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.5.4.1. Request

This table shows the URI parameters for the live-migrate server (os-migrateLive action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.63. Live-migrate server (os-migrateLive action): JSON request

```
{
  "os-migrateLive": {
    "host": "01c0cadedf72d47e28a672a76060d492c",
    "block_migration": false,
    "disk_over_commit": false
  }
}
```

4.5.5. Reset networking on a server (resetNetwork action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resets networking on a server.

Specify the resetNetwork action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.5.5.1. Request

This table shows the URI parameters for the reset networking on a server (resetnetwork action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.64. Reset networking on a server (resetNetwork action): JSON request

```
{
  "resetNetwork": null
}
```

4.5.6. Reset server state (os-resetState action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Resets the state of a server.

Specify the `os-resetState` action and the state in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.5.6.1. Request

This table shows the URI parameters for the reset server state (`os-resetstate` action) request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.65. Reset server state (os-resetState action): JSON request

```
{
  "os-resetState": {
    "state": "active"
  }
}
```

4.6. Servers diagnostics (servers, diagnostics)

Shows the usage data for a server.

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/diagnostics	Shows basic usage data for a server.

4.6.1. Show server diagnostics

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/diagnostics	Shows basic usage data for a server.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.6.1.1. Request

This table shows the URI parameters for the show server diagnostics request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.6.1.2. Response

Example 4.66. Server diagnostics: JSON response

```
{
  "cpu0_time": 173000000000,
  "memory": 524288,
  "vda_errors": -1,
  "vda_read": 262144,
  "vda_read_req": 112,
  "vda_write": 5778432,
  "vda_write_req": 488,
  "vnet1_rx": 2070139,
  "vnet1_rx_drop": 0,
  "vnet1_rx_errors": 0,
  "vnet1_rx_packets": 26701,
  "vnet1_tx": 140208,
  "vnet1_tx_drop": 0,
  "vnet1_tx_errors": 0,
  "vnet1_tx_packets": 662
}
```

4.7. Servers IPs (servers, ips)

Lists the IP addresses for an instance and shows details for an IP address.

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/ips	Lists IP addresses that are assigned to an instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/ips/{network_label}	Shows IP addresses details for a network label of a server instance.

4.7.1. List IPs

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/ips	Lists IP addresses that are assigned to an instance.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.7.1.1. Request

This table shows the URI parameters for the list ips request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.7.1.2. Response

Example 4.67. List IPs: JSON response

```
{
  "addresses": {
    "private": [
      {
        "addr": "192.168.0.3",
        "version": 4
      }
    ]
  }
}
```

4.7.2. Show IP details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/ips/{network_label}	Shows IP addresses details for a network label of a server instance.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.7.2.1. Request

This table shows the URI parameters for the show ip details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{network_label}	String	The network label, such as public or private.

This operation does not accept a request body.

4.7.2.2. Response

Example 4.68. Show IP details: JSON response

```
{
  "private": [
    {
      "addr": "192.168.0.3",
      "version": 4
    }
  ]
}
```

4.8. Server metadata (servers, metadata)

Lists metadata, creates or replaces one or more metadata items, and updates one or more metadata items for a server.

Shows details for, creates or replaces, and updates a metadata item, by key, for a server.

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/metadata	Lists all metadata for a server.
PUT	/v2.1/{tenant_id}/servers/{server_id}/metadata	Creates or replaces one or more metadata items for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/metadata	Updates one or more metadata items for a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}	Shows details for a metadata item, by key, for a server.

Method	URI	Description
PUT	/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}	Creates or replaces a metadata item, by key, for a server.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}	Deletes a metadata item, by key, from a server.

4.8.1. List all metadata

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/metadata	Lists all metadata for a server.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.8.1.1. Request

This table shows the URI parameters for the list all metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.8.1.2. Response

Example 4.69. List all metadata: JSON response

```
{
  "metadata": {
    "foo": "Foo Value"
  }
}
```

4.8.2. Create or replace metadata items

Method	URI	Description
PUT	/v2.1/{tenant_id}/servers/{server_id}/metadata	Creates or replaces one or more metadata items for a server.

Creates any metadata items that do not already exist in the server. Removes and completely replaces any metadata items that already exist in the server with the metadata items in the request.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.8.2.1. Request

This table shows the URI parameters for the create or replace metadata items request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.70. Create or replace metadata items: JSON request

```
{
  "metadata": {
    "foo": "Foo Value"
  }
}
```

4.8.2.2. Response

Example 4.71. Create or replace metadata items: JSON response

```
{
  "metadata": {
    "foo": "Foo Value"
  }
}
```

4.8.3. Update metadata items

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/metadata	Updates one or more metadata items for a server.

Replaces metadata items that match keys. Does not modify items that are not in the request.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.8.3.1. Request

This table shows the URI parameters for the update metadata items request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.72. Update metadata items: JSON request

```
{
  "metadata": {
    "foo": "Foo Value"
  }
}
```

4.8.3.2. Response

Example 4.73. Update metadata items: JSON response

```
{
  "metadata": {
    "foo": "Foo Value"
  }
}
```

4.8.4. Show metadata item details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}	Shows details for a metadata item, by key, for a server.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.8.4.1. Request

This table shows the URI parameters for the show metadata item details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{key}	String	The metadata item key, as a string. Maximum length is 255 characters.

This operation does not accept a request body.

4.8.4.2. Response

Example 4.74. Show metadata item details: JSON response

```
{
  "meta": {
    "foo": "Foo Value"
  }
}
```

4.8.5. Create or update metadata item

Method	URI	Description
PUT	/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}	Creates or replaces a metadata item, by key, for a server.

Creates a metadata item that does not already exist in the server. Removes and completely replaces a metadata item that already exists in the server with the metadata item in the request.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.8.5.1. Request

This table shows the URI parameters for the create or update metadata item request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{key}	String	The metadata item key, as a string. Maximum length is 255 characters.

Example 4.75. Create or update metadata item: JSON request

```
{
  "meta": {
    "foo": "Bar Value"
  }
}
```

4.8.5.2. Response

Example 4.76. Create or update metadata item: JSON response

```
{
  "meta": {
    "foo": "Bar Value"
  }
}
```

4.8.6. Delete metadata item

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}	Deletes a metadata item, by key, from a server.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 204

4.8.6.1. Request

This table shows the URI parameters for the delete metadata item request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{key}	String	The metadata item key, as a string. Maximum length is 255 characters.

This operation does not accept a request body.

4.9. Servers - list actions (servers, os-instance-actions)

All users can list available actions for a server.

All users can show details for a server action; however, only administrators can view server events in server action details.

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-instance-actions	Lists actions for a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-instance-actions/{request_id}	Shows details for a server action.

4.9.1. List actions for server

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-instance-actions	Lists actions for a server.

Normal response codes: 200

4.9.1.1. Request

This table shows the URI parameters for the list actions for server request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.9.1.2. Response

Example 4.77. List actions for server: JSON response

```
{
  "instanceActions": [
    {
      "action": "resize",
      "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
      "message": "",
      "project_id": "842",
      "request_id": "req-25517360-b757-47d3-be45-0e8d2a01b36a",
      "start_time": "2012-12-05T01:00:00.000000",
      "user_id": "789"
    },
    {
      "action": "reboot",
      "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
      "message": "",
      "project_id": "147",
      "request_id": "req-3293a3f1-b44c-4609-b8d2-d81b105636b8",
      "start_time": "2012-12-05T00:00:00.000000",
      "user_id": "789"
    }
  ]
}
```


4.9.2. Show server action details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-instance-actions/{request_id}	Shows details for a server action.

Normal response codes: 200

4.9.2.1. Request

This table shows the URI parameters for the show server action details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{request_id}	String	The ID of the request.

This operation does not accept a request body.

4.9.2.2. Response

Example 4.78. Show server action details: JSON response

```
{
  "instanceAction": {
    "action": "reboot",
    "events": [
      {
        "event": "schedule",
        "finish_time": "2012-12-05T01:02:00.000000",
        "result": "Success",
        "start_time": "2012-12-05T01:00:02.000000",
        "traceback": ""
      },
      {
        "event": "compute_create",
        "finish_time": "2012-12-05T01:04:00.000000",
        "result": "Success",
        "start_time": "2012-12-05T01:03:00.000000",
        "traceback": ""
      }
    ],
    "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
    "message": "",
    "project_id": "147",
    "request_id": "req-3293a3f1-b44c-4609-b8d2-d81b105636b8",
    "start_time": "2012-12-05T00:00:00.000000",
    "user_id": "789"
  }
}
```

4.10. Port interfaces (servers, os-interface)

Creates a port interface and uses it to attach a port to a server and detaches a port interface from a server. Also, lists all port interfaces and shows details for a port interface.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-interface	Creates a port interface and uses it to attach a port to a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-interface	Lists port interfaces that are attached to a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-interface/{port_id}	Shows details for a port interface that is attached to a server.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-interface/{port_id}	Detaches a port interface.

4.10.1. Create interface

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-interface	Creates a port interface and uses it to attach a port to a server instance.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), badMediaType (415), NetworkNotFound (400)

4.10.1.1. Request

This table shows the URI parameters for the create interface request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.79. Create interface: JSON request

```
{
  "interfaceAttachment": {
    "port_id": "ce531f90-199f-48c0-816c-13e38010b442"
  }
}
```

4.10.1.2. Response

Example 4.80. Create interface: JSON response

```
{
  "interfaceAttachment": {
    "fixed_ips": [
      {
        "ip_address": "192.168.1.3",
        "subnet_id": "f8a6e8f8-c2ec-497c-9f23-da9616de54ef"
      }
    ],
    "mac_addr": "fa:16:3e:4c:2c:30",
    "net_id": "3cb9bc59-5699-4588-a4b1-b87f96708bc6",
    "port_id": "ce531f90-199f-48c0-816c-13e38010b442",
    "port_state": "ACTIVE"
  }
}
```

4.10.2. List port interfaces

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-interface	Lists port interfaces that are attached to a server.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.10.2.1. Request

This table shows the URI parameters for the list port interfaces request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.10.2.2. Response

Example 4.81. List port interfaces: JSON response

```
{
  "interfaceAttachments": [
    {
      "fixed_ips": [
        {
          "ip_address": "192.168.1.3",
          "subnet_id": "f8a6e8f8-c2ec-497c-9f23-da9616de54ef"
        }
      ],
      "mac_addr": "fa:16:3e:4c:2c:30",
      "net_id": "3cb9bc59-5699-4588-a4b1-b87f96708bc6",
      "port_id": "ce531f90-199f-48c0-816c-13e38010b442",
      "port_state": "ACTIVE"
    }
  ]
}
```

4.10.3. Show port interface details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-interface/{port_id}	Shows details for a port interface that is attached to a server.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.10.3.1. Request

This table shows the URI parameters for the show port interface details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{port_id}	UUID	The UUID of the port.

This operation does not accept a request body.

4.10.3.2. Response

Example 4.82. Show port interface details: JSON response

```
{
  "interfaceAttachment": {
    "fixed_ips": [
      {
        "ip_address": "192.168.1.3",
        "subnet_id": "f8a6e8f8-c2ec-497c-9f23-da9616de54ef"
      }
    ],
    "mac_addr": "fa:16:3e:4c:2c:30",
    "net_id": "3cb9bc59-5699-4588-a4b1-b87f96708bc6",
    "port_id": "ce531f90-199f-48c0-816c-13e38010b442",
    "port_state": "ACTIVE"
  }
}
```

4.10.4. Detach interface

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-interface/{port_id}	Detaches a port interface.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), badMediaType (415), NetworkNotFound (400)

4.10.4.1. Request

This table shows the URI parameters for the detach interface request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{port_id}	UUID	The UUID of the port.

This operation does not accept a request body.

4.11. Servers password (servers, os-server-password)

Shows the encrypted administrative password. Also, clears the encrypted administrative password for a server, which removes it from the metadata server.

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-server-password	Shows the administrative password for a server.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-server-password	Clears the encrypted administrative password for a server, which removes it from the metadata server.

4.11.1. Show server password

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-server-password	Shows the administrative password for a server.

This operation calls the metadata service to query metadata information and does not read password information from the server itself.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.11.1.1. Request

This table shows the URI parameters for the show server password request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.11.1.2. Response

Example 4.83. Show server password: JSON response

```
{
  "password": "xloz03wLCBRWaa2yDjCCVx8vwNPypxnypmRYDa/zEr1Q+EzPe1S/
Gz6nfmC52m0l0SCRuU0mG7kqqgejPof6M7b0ezS387zjq4LSvvp28zUknzy4YzfFGhnHAdai3TxUJ26pfQCYrq8UTzm
I1K2LsuipfxSJR7Wdke4zNXJjHHP2RfYsVbZ/k9ANu+Nz4iIH8/7Cacud/
pphH7EjrY6a4RZnrjQskrhKYed0YERpotyjYk1eDtRe72GrSiXteqCM4biaQ5w3ruS+AcX//
PXk3uJ5kC7d67fPXaVz4WaQRYMg=="
}
```

4.11.2. Clear admin password

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-server-password	Clears the encrypted administrative password for a server, which removes it from the metadata server.

This action does not actually change the instance server password.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 204

4.11.2.1. Request

This table shows the URI parameters for the clear admin password request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.12. Servers virtual interfaces (servers, os-virtual-interfaces)

Lists virtual interfaces for a server instance.

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Lists the virtual interfaces for an instance.

4.12.1. List virtual interfaces

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Lists the virtual interfaces for an instance.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Change these permissions through the `policy.json` file.

The API v2 returns the network ID in the `OS-EXT-VIF-NET:net_id` response attribute.

The API v2.1 base version does not return the network ID.

The API v2.12 microversion returns the network ID in the `net_id` response attribute.

Normal response codes: 200

Error response codes: `computeFault` (400, 500, ...), `serviceUnavailable` (503), `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `itemNotFound` (404)

4.12.1.1. Request

This table shows the URI parameters for the list virtual interfaces request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.12.1.2. Response

Example 4.84. List virtual interfaces: JSON response

```
{
  "virtual_interfaces": [
    {
      "id": "cec8b9bb-5d22-4104-b3c8-4c35db3210a6",
      "mac_address": "fa:16:3e:3c:ce:6f"
    }
  ]
}
```

4.13. Servers with volume attachments (servers, os-volume_attachments)

Attaches volumes that are created through the volume API to server instances. Also, lists volume attachments for a server instance, shows details for a volume attachment, and detaches a volume.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments	Attaches a volume to a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments	Lists the volume attachments for a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Shows details for a volume attachment.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Detaches a volume from a server.

4.13.1. Attach volume to server

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments	Attaches a volume to a server.

Preconditions

- You can attach a volume to a server with an ACTIVE, PAUSED, SHUTOFF, VERIFY_RESIZE, or SOFT_DELETED status.
- You can attach a volume to a server with a status that is not locked.
- You can attach a volume with a available status.
- You can attach a volume when its volume type supports volume attaching.

Asynchronous Postconditions

- After successfully attaching a volume to a server, the volume status changes to in-use.
- You can see the attached volume when you log in to the server.

Troubleshooting

- If the volume status remains in available state, the request failed. Ensure that you meet the preconditions and run the request again. If the request fails again, investigate the server and the volume status.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), bad-MediaType (415), NetworkNotFound (400), buildInProgress (409)

4.13.1.1. Request

This table shows the URI parameters for the attach volume to server request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.85. Attach volume to server: JSON request

```
{
  "volumeAttachment": {
    "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
    "device": "/dev/vdd"
  }
}
```

4.13.1.2. Response

Example 4.86. Attach volume to server: JSON response

```
{
  "volumeAttachment": {
    "device": "/dev/vdd",
    "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
    "serverId": "0c92f3f6-c253-4c9b-bd43-e880a8d2eb0a",
    "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
  }
}
```

4.13.2. List volume attachments

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments	Lists the volume attachments for a server.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.13.2.1. Request

This table shows the URI parameters for the list volume attachments request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.13.2.2. Response

Example 4.87. List volume attachments: JSON response

```
{
  "volumeAttachments": [
    {
      "device": "/dev/sdd",
      "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
      "serverId": "4d8c3732-a248-40ed-bebc-539a6ffd25c0",
      "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
    },
    {
      "device": "/dev/sdc",
      "id": "a26887c6-c47b-4654-abb5-dfadf7d3f804",
      "serverId": "4d8c3732-a248-40ed-bebc-539a6ffd25c0",
      "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f804"
    }
  ]
}
```

4.13.3. Show volume attachment details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Shows details for a volume attachment.

Preconditions

- The volume must be attached to the server.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.13.3.1. Request

This table shows the URI parameters for the show volume attachment details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{attachment_id}	UUID	The UUID of the attachment.

This operation does not accept a request body.

4.13.3.2. Response

Example 4.88. Show volume attachment details: JSON response

```
{
  "volumeAttachment": {
    "device": "/dev/sdd",
    "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
    "serverId": "2390fb4d-1693-45d7-b309-e29c4af16538",
    "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
  }
}
```

4.13.4. Detach volume

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Detaches a volume from a server.

Preconditions

- The volume must be attached to the server.
- You can detach a volume when the server status is ACTIVE, PAUSED, SHUTOFF, VERIFY_RESIZE, or SOFT_DELETED.
- You can detach a volume when its status is in-use.
- You can detach a volume from a server with a status that is not locked.
- You cannot detach a root device volume.

Asynchronous Postconditions

- After successfully detaching a volume, its status changes to available.
- The detached volume is no longer visible on the server.

Troubleshooting

- If the volume status remains in in-use state, the request failed. Ensure that you meet the preconditions and run the request again. If the request fails again, investigate whether another operation is running that causes a race condition.
- If you attempt to detach a root device volume, this operation returns the Forbidden (403) response code.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.13.4.1. Request

This table shows the URI parameters for the detach volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{attachment_id}	UUID	The UUID of the attachment.

This operation does not accept a request body.

4.14. Flavors with extended attributes (flavors)

Shows information about flavors.

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors{?minDisk,minRam,sort_key,sort_dir,limit,marker}	Lists flavors.
POST	/v2.1/{tenant_id}/flavors	Creates a flavor.
GET	/v2.1/{tenant_id}/flavors/{flavor_id}	Shows details for a flavor.
DELETE	/v2.1/{tenant_id}/flavors/{flavor_id}	Deletes a flavor.
GET	/v2.1/{tenant_id}/flavors/detail{?minDisk,minRam,sort_key,sort_dir,limit,marker}	Lists flavors with details.

4.14.1. List flavors

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors{?minDisk,minRam,sort_key,sort_dir,limit,marker}	Lists flavors.

Normal response codes: 200

4.14.1.1. Request

This table shows the URI parameters for the list flavors request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list flavors request:

Name	Type	Description
minDisk	Int (Optional)	Filters the response by a minimum disk space, in GB.
minRam	Int (Optional)	Filters the response by a minimum RAM, in MB.
sort_key	String (Optional)	Sorts by a server attribute. Default attribute is created. You can specify multiple pairs of sort key and sort direction query parameters. If you omit the sort direction in a pair, the API uses the natural sorting direction of the server sort_key attribute.
sort_dir	String (Optional)	Sort direction. A valid value is asc (ascending) or desc (descending). Default is desc. You can specify multiple pairs of sort key and sort direction query parameters. If you omit the sort direction in a pair, the API uses the natural sorting direction of the direction of the server sort_key attribute.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

4.14.1.2. Response

Example 4.89. List flavors: JSON response

```
{
  "flavors": [
    {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/v2.1/openstack/
flavors/1",
          "rel": "self"
        }
      ]
    }
  ]
}
```

```

        },
        {
            "href": "http://openstack.example.com/openstack/flavors/
1",
            "rel": "bookmark"
        }
    ],
    "name": "m1.tiny"
},
{
    "id": "2",
    "links": [
        {
            "href": "http://openstack.example.com/v2.1/openstack/
flavors/2",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/flavors/
2",
            "rel": "bookmark"
        }
    ],
    "name": "m1.small"
},
{
    "id": "3",
    "links": [
        {
            "href": "http://openstack.example.com/v2.1/openstack/
flavors/3",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/flavors/
3",
            "rel": "bookmark"
        }
    ],
    "name": "m1.medium"
},
{
    "id": "4",
    "links": [
        {
            "href": "http://openstack.example.com/v2.1/openstack/
flavors/4",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/flavors/
4",
            "rel": "bookmark"
        }
    ],
    "name": "m1.large"
},
{
    "id": "5",
    "links": [

```

```
flavors/5",
    {
        "href": "http://openstack.example.com/v2.1/openstack/
    },
    {
        "href": "http://openstack.example.com/openstack/flavors/
    },
    {
        "name": "m1.xlarge"
    }
]
```

4.14.2. Create flavor

Method	URI	Description
POST	/v2.1/{tenant_id}/flavors	Creates a flavor.

Normal response codes: 200

4.14.2.1. Request

This table shows the URI parameters for the create flavor request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.90. Create flavor: JSON request

```
{
  "flavor": {
    "name": "test_flavor",
    "ram": 1024,
    "vcpus": 2,
    "disk": 10,
    "id": "10"
  }
}
```

4.14.2.2. Response

Example 4.91. Create flavor: JSON response

```
{
  "flavor": {
    "OS-FLV-DISABLED:disabled": false,
    "disk": 10,
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "os-flavor-access:is_public": true,
    "id": "10",
    "links": [
      {
        "href": "http://openstack.example.com/v2.1/flavors/10",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/flavors/10",
        "rel": "bookmark"
      }
    ],
    "name": "test_flavor",
    "ram": 1024,
    "swap": "",
    "vcpus": 2
  }
}
```

4.14.3. Show flavor details

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/{flavor_id}	Shows details for a flavor.

Normal response codes: 200

4.14.3.1. Request

This table shows the URI parameters for the show flavor details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{flavor_id}	UUID	The ID of the flavor.

4.14.3.2. Response

Example 4.92. Show flavor details: JSON response

```
{
  "flavor": {
    "OS-FLV-DISABLED:disabled": false,
    "disk": 1,
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "os-flavor-access:is_public": true,
    "id": "1",
    "links": [
      {
        "href": "http://openstack.example.com/v2.1/openstack/flavors/1",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/1",
        "rel": "bookmark"
      }
    ],
    "name": "m1.tiny",
    "ram": 512,
    "swap": "",
    "vcpus": 1
  }
}
```

4.14.4. Delete flavor

Method	URI	Description
DELETE	/v2.1/{tenant_id}/flavors/{flavor_id}	Deletes a flavor.

Normal response codes: 202

4.14.4.1. Request

This table shows the URI parameters for the delete flavor request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{flavor_id}	UUID	The ID of the flavor.

This operation does not accept a request body.

4.14.5. List flavors with details

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/detail{? minDisk,minRam,sort_key,sort_dir, limit,marker}	Lists flavors with details.

Normal response codes: 200

4.14.5.1. Request

This table shows the URI parameters for the list flavors with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list flavors with details request:

Name	Type	Description
minDisk	Int (Optional)	Filters the response by a minimum disk space, in GB.
minRam	Int (Optional)	Filters the response by a minimum RAM, in MB.
sort_key	String (Optional)	Sorts by a server attribute. Default attribute is created. You can specify multiple pairs of sort key and sort direction query parameters. If you omit the sort direction in a pair, the API uses the natural sorting direction of the server sort_key attribute.
sort_dir	String (Optional)	Sort direction. A valid value is asc (ascending) or desc (descending). Default is desc. You can specify multiple pairs of sort key and sort direction query parameters. If you omit the sort direction in a pair, the API uses the natural sorting direction of the direction of the server sort_key attribute.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

4.14.5.2. Response

Example 4.93. List flavors with details: JSON response

```
{
  "flavors": [
    {
      "OS-FLV-DISABLED:disabled": false,
      "disk": 1,
      "OS-FLV-EXT-DATA:ephemeral": 0,
      "os-flavor-access:is_public": true,
      "id": "1",
      "links": [
```

```

        {
            "href": "http://openstack.example.com/v2.1/openstack/
flavors/1",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/flavors/
1",
            "rel": "bookmark"
        }
    ],
    "name": "m1.tiny",
    "ram": 512,
    "swap": "",
    "vcpus": 1
},
{
    "OS-FLV-DISABLED:disabled": false,
    "disk": 20,
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "os-flavor-access:is_public": true,
    "id": "2",
    "links": [
        {
            "href": "http://openstack.example.com/v2.1/openstack/
flavors/2",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/flavors/
2",
            "rel": "bookmark"
        }
    ],
    "name": "m1.small",
    "ram": 2048,
    "swap": "",
    "vcpus": 1
},
{
    "OS-FLV-DISABLED:disabled": false,
    "disk": 40,
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "os-flavor-access:is_public": true,
    "id": "3",
    "links": [
        {
            "href": "http://openstack.example.com/v2.1/openstack/
flavors/3",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/flavors/
3",
            "rel": "bookmark"
        }
    ],
    "name": "m1.medium",
    "ram": 4096,
    "swap": "",

```



```

        "vcpus": 2
      },
      {
        "OS-FLV-DISABLED:disabled": false,
        "disk": 80,
        "OS-FLV-EXT-DATA:ephemeral": 0,
        "os-flavor-access:is_public": true,
        "id": "4",
        "links": [
          {
            "href": "http://openstack.example.com/v2.1/openstack/
flavors/4",
            "rel": "self"
          },
          {
            "href": "http://openstack.example.com/openstack/flavors/
4",
            "rel": "bookmark"
          }
        ],
        "name": "m1.large",
        "ram": 8192,
        "swap": "",
        "vcpus": 4
      },
      {
        "OS-FLV-DISABLED:disabled": false,
        "disk": 160,
        "OS-FLV-EXT-DATA:ephemeral": 0,
        "os-flavor-access:is_public": true,
        "id": "5",
        "links": [
          {
            "href": "http://openstack.example.com/v2.1/openstack/
flavors/5",
            "rel": "self"
          },
          {
            "href": "http://openstack.example.com/openstack/flavors/
5",
            "rel": "bookmark"
          }
        ],
        "name": "m1.xlarge",
        "ram": 16384,
        "swap": "",
        "vcpus": 8
      }
    ]
  }
}

```

4.15. Flavors access (flavors, os-flavor-access, action)

Lists tenants who have access to a private flavor and adds private flavor access to and removes private flavor access from tenants. By default, only administrators can manage pri-

vate flavor access. A private flavor has `is_public` set to `false` while a public flavor has `is_public` set to `true`.

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/os-flavor-access	Lists flavor access information.
POST	/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action	Adds flavor access to a tenant and flavor.
POST	/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action	Removes flavor access from a tenant and flavor.

4.15.1. List flavor access information for given flavor

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/os-flavor-access	Lists flavor access information.

Normal response codes: 200

4.15.1.1. Request

This table shows the URI parameters for the list flavor access information for given flavor request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.15.1.2. Response

Example 4.94. List flavor access information for given flavor: JSON response

```
{
  "flavor_access": [
    {
      "flavor_id": "10",
      "tenant_id": "fake_tenant"
    }
  ]
}
```

4.15.2. Add flavor access to tenant

Method	URI	Description
POST	/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action	Adds flavor access to a tenant and flavor.

Specify the addTenantAccess action and the tenant_id in the request body.

Normal response codes: 200

4.15.2.1. Request

This table shows the URI parameters for the add flavor access to tenant request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{flavor_id}	UUID	The ID of the flavor.

Example 4.95. Add flavor access to tenant: JSON request

```
{
  "addTenantAccess": {
    "tenant": "fake_tenant"
  }
}
```

4.15.2.2. Response

Example 4.96. Add flavor access to tenant: JSON response

```
{
  "flavor_access": [
    {
      "flavor_id": "10",
      "tenant_id": "fake_tenant"
    }
  ]
}
```

4.15.3. Remove flavor access from tenant

Method	URI	Description
POST	/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action	Removes flavor access from a tenant and flavor.

Specify the `removeTenantAccess` action and the `tenant_id` in the request body.

Normal response codes: 200

4.15.3.1. Request

This table shows the URI parameters for the remove flavor access from tenant request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{flavor_id}	UUID	The ID of the flavor.

Example 4.97. Remove flavor access from tenant: JSON request

```
{
  "removeTenantAccess": {
    "tenant": "fake_tenant"
  }
}
```

4.15.3.2. Response

Example 4.98. Remove flavor access from tenant: JSON response

```
{
  "flavor_access": []
}
```

4.16. Flavors extra-specs (flavors, os-flavor-extra-specs)

Lists, creates, deletes, and updates the extra-specs or keys for a flavor.

Method	URI	Description
POST	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Creates extra specs for a flavor, by ID.
GET	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Lists all extra specs for a flavor, by ID.
GET	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}	Shows an extra spec, by key, for a flavor, by ID.
PUT	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}	Updates an extra spec, by key, for a flavor, by ID.

Method	URI	Description
DELETE	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}	Deletes an extra spec, by key, for a flavor, by ID.

4.16.1. Create extra specs for a flavor

Method	URI	Description
POST	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Creates extra specs for a flavor, by ID.

Normal response codes: 200

4.16.1.1. Request

This table shows the URI parameters for the create extra specs for a flavor request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{flavor_id}	UUID	The ID of the flavor.

Example 4.99. Create extra specs for a flavor: JSON request

```
{
  "extra_specs": {
    "key1": "value1",
    "key2": "value2"
  }
}
```

4.16.1.2. Response

Example 4.100. Create extra specs for a flavor: JSON response

```
{
  "extra_specs": {
    "key1": "value1",
    "key2": "value2"
  }
}
```

4.16.2. List extra specs for a flavor

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Lists all extra specs for a flavor, by ID.

Normal response codes: 200

4.16.2.1. Request

This table shows the URI parameters for the list extra specs for a flavor request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{flavor_id}	UUID	The ID of the flavor.

Example 4.101. List extra specs for a flavor: JSON request

```
{
  "extra_specs": {
    "key1": "value1",
    "key2": "value2"
  }
}
```

4.16.2.2. Response

Example 4.102. List extra specs for a flavor: JSON response

```
{
  "extra_specs": {
    "key1": "value1",
    "key2": "value2"
  }
}
```


4.16.3. Show an extra spec for a flavor

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}	Shows an extra spec, by key, for a flavor, by ID.

Normal response codes: 200

4.16.3.1. Request

This table shows the URI parameters for the show an extra spec for a flavor request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{flavor_id}	UUID	The ID of the flavor.
{flavor_extra_spec_key}	String	The extra spec key for the flavor.

This operation does not accept a request body.

4.16.3.2. Response

Example 4.103. Show an extra spec for a flavor: JSON response

```
{
  "key1": "value1"
}
```

4.16.4. Update an extra spec for a flavor

Method	URI	Description
PUT	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}	Updates an extra spec, by key, for a flavor, by ID.

Normal response codes: 200

4.16.4.1. Request

This table shows the URI parameters for the update an extra spec for a flavor request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{flavor_id}	UUID	The ID of the flavor.
{flavor_extra_spec_key}	String	The extra spec key for the flavor.

Example 4.104. Update an extra spec for a flavor: JSON request

```
{
  "key1": "new_value1"
}
```

4.16.4.2. Response

Example 4.105. Update an extra spec for a flavor: JSON response

```
{
  "key1": "new_value1"
}
```

4.16.5. Delete an extra spec for a flavor

Method	URI	Description
DELETE	/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}	Deletes an extra spec, by key, for a flavor, by ID.

Normal response codes: 200

4.16.5.1. Request

This table shows the URI parameters for the delete an extra spec for a flavor request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{flavor_id}	UUID	The ID of the flavor.
{flavor_extra_spec_key}	String	The extra spec key for the flavor.

This operation does not accept a request body.

4.17. Keypairs (keypairs)

Generates, imports, and deletes SSH keys.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-keypairs	Lists keypairs that are associated with the account.
POST	/v2.1/{tenant_id}/os-keypairs	Generates or imports a keypair.
DELETE	/v2.1/{tenant_id}/os-keypairs/{keypair_name}	Deletes a keypair.
GET	/v2.1/{tenant_id}/os-keypairs/{keypair_name}	Shows details for a keypair that is associated with the account.

4.17.1. List keypairs

Method	URI	Description
GET	/v2.1/{tenant_id}/os-keypairs	Lists keypairs that are associated with the account.

Normal response codes: 200

4.17.1.1. Request

This table shows the URI parameters for the list keypairs request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.17.1.2. Response

Example 4.106. List keypairs: JSON response

```
{
  "keypairs": [
    {
      "keypair": {
        "fingerprint":
"7e:eb:ab:24:ba:d1:e1:88:ae:9a:fb:66:53:df:d3:bd",
        "name": "keypair-50ca852e-273f-4cdc-8949-45feba200837",
        "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCKF3MX590r1Bs3dH5CU7lNmvpbrgZxSpyGjlnE8Flkirnc/
Up22lpjznnoxqeoTAwTW034k7Dz6aYIrZGmQwe2Tke084yqvlj45Dkyoj95fw/
sZacm0cZNuL69E0bEGHdprfGJQajrpz22NQoCD8TFB8Wv+8om9NH9Le6s
+WPe98WC77KLw8qgfQsbIey+JawPwL4067ZdL5xrypuRjfIPWjgy/
VH85IXg/Z/GONZ2nxHgSShMkwqSFECAC5L3PHB+0+/12M/
iikdatFSVGjpuHvkL0s3oe7m6Hl0fluSJ85BzLWBbvva93qkGmLg4ZAc8rPh20+YIsBUHNLLMM/oQp
Generated-by-Nova\n"
      }
    }
  ]
}
```

4.17.2. Create or import keypair

Method	URI	Description
POST	/v2.1/{tenant_id}/os-keypairs	Generates or imports a keypair.

Normal response codes: 200

4.17.2.1. Request

This table shows the URI parameters for the create or import keypair request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.107. Create or import keypair: JSON request

```
{
  "keypair": {
    "name": "keypair-d20a3d59-9433-4b79-8726-20b431d89c78",
    "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDx8nkQv/
zgGgB4rMYmIf+6A4l6Rr+o/6lHBQdW5aYd44bd8JttDCE/F/pNRr0lRE
+PiqSP08nDPHw0010JeMH9gYgnnFlyY3/OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/
0d3ac30bNH6Sw9JD9UZHycpSxsIbECHw== Generated-by-Nova"
  }
}
```

4.17.2.2. Response

Example 4.108. Create or import keypair: JSON response

```
{
  "keypair": {
    "fingerprint": "1e:2c:9b:56:79:4b:45:77:f9:ca:7a:98:2c:b0:d5:3c",
    "name": "keypair-803a1926-af78-4b05-902a-1d6f7a8d9d3e",
    "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDx8nkQv/
zgGgB4rMYmIf+6A4l6Rr+o/6lHBQdW5aYd44bd8JttDCE/F/pNRr0lRE
+PiqSP08nDPHw0010JeMH9gYgnnFlyY3/OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/
0d3ac30bNH6Sw9JD9UZHycpSxsIbECHw== Generated-by-Nova",
    "user_id": "fake"
  }
}
```

4.17.3. Delete keypair

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-key-pairs/{keypair_name}	Deletes a keypair.

Normal response codes: 204

4.17.3.1. Request

This table shows the URI parameters for the delete keypair request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{keypair_name}	String	The keypair name.

This operation does not accept a request body.

4.17.4. Show keypair details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-key-pairs/{keypair_name}	Shows details for a keypair that is associated with the account.

Normal response codes: 200

4.17.4.1. Request

This table shows the URI parameters for the show keypair details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{keypair_name}	String	The keypair name.

This operation does not accept a request body.

4.17.4.2. Response

Example 4.109. Show keypair details: JSON response

```
{
  "keypair": {
    "fingerprint": "44:fe:29:6e:23:14:b9:53:5b:65:82:58:1c:fe:5a:c3",
    "name": "keypair-6638abdb-c4e8-407c-ba88-c8dd7cc3c4f1",
    "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQAC1HTrHCbb9NawNLSV8N6tSa8i637+EC2dA+lsdHHfQlT54t
+N0nHhJPlKWDLhc579j87vp6RDFriFJ/smsTnDnf64012z0kBaJpJPH2zXrBkZFK6q2rmxydURzX/
z0yLSCP77SFJ0fdXWH2hMsAusflGyryHGx20n
+mZK6mDrxVzGxEz228dwQ5G7Az50oZDwygH2pqPvKjkifRw0jwUKf3BbkP0QvANAC0k26cv16mNFpFJfI1N30C5lUsZQ
qup58J5kflNm7I61sy1mJon6SGqNUSfoQagqtBH6vd/tU1jnIwZ03uUroAL Generated-by-Nova\
n",
    "user_id": "fake",
    "deleted": false,
    "created_at": "2014-05-07T12:06:13.681238",
    "updated_at": null,
    "deleted_at": null,
    "id": 1
  }
}
```

4.18. Limits (limits)

Shows rate and absolute limits for the tenant.

Method	URI	Description
GET	/v2.1/{tenant_id}/limits	Shows rate and absolute limits for the tenant.

4.18.1. Show rate and absolute limits

Method	URI	Description
GET	/v2.1/{tenant_id}/limits	Shows rate and absolute limits for the tenant.

Normal response codes: 200

4.18.1.1. Request

This table shows the URI parameters for the show rate and absolute limits request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.18.1.2. Response

Example 4.110. Show rate and absolute limits: JSON response

```
{
  "limits": {
    "rate": [],
    "absolute": {
      "maxServerMeta": 128,
      "maxPersonality": 5,
      "totalServerGroupsUsed": 0,
      "maxImageMeta": 128,
      "maxPersonalitySize": 10240,
      "maxTotalKeypairs": 100,
      "maxSecurityGroupRules": 20,
      "maxServerGroups": 10,
      "totalCoresUsed": 1,
      "totalRAMUsed": 2048,
      "totalInstancesUsed": 1,
      "maxSecurityGroups": 10,
      "totalFloatingIpsUsed": 0,
      "maxTotalCores": 20,
      "maxServerGroupMembers": 10,
      "maxTotalFloatingIps": 10,
      "totalSecurityGroupsUsed": 1,
      "maxTotalInstances": 10,
      "maxTotalRAMSize": 51200
    }
  }
}
```

4.19. Extensions (extensions)

Lists available extensions and shows information for an extension, by alias.

OpenStack Compute API v2.0 supports extensions. However, extensions are deprecated in Compute API v2.1. For information about extensions, see [Extensions](#).

Method	URI	Description
GET	/v2.1/{tenant_id}/extensions	Lists available extensions.
GET	/v2.1/{tenant_id}/extensions/{alias}	Shows details for an extension, by alias.

4.19.1. List extensions

Method	URI	Description
GET	/v2.1/{tenant_id}/extensions	Lists available extensions.

Extensions introduce features and vendor-specific functionality to the API without requiring a version change.

The response shows the extension name and its alias. To show details for an extension, you specify the alias.

Normal response codes: 200203

4.19.1.1. Request

This table shows the URI parameters for the list extensions request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.19.1.2. Response

Example 4.111. List extensions: JSON response

```
{
  "extensions": [
    {
      "alias": "NMN",
      "description": "Multiple network support.",
      "links": [],
      "name": "Multinic",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "OS-DCF",
      "description": "Disk Management Extension.",
      "links": [],
      "name": "DiskConfig",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "OS-EXT-AZ",
      "description": "Extended Availability Zone support.",
      "links": [],
      "name": "ExtendedAvailabilityZone",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "OS-EXT-IMG-SIZE",
      "description": "Adds image size to image listings.",
      "links": [],
      "name": "ImageSize",
```

```

    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "OS-EXT-IPS",
    "description": "",
    "links": [],
    "name": "ExtendedIps",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "OS-EXT-IPS-MAC",
    "description": "",
    "links": [],
    "name": "ExtendedIpsMac",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "OS-EXT-SRV-ATTR",
    "description": "Extended Server Attributes support.",
    "links": [],
    "name": "ExtendedServerAttributes",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "OS-EXT-STS",
    "description": "",
    "links": [],
    "name": "ExtendedStatus",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "OS-EXT-VIF-NET",
    "description": "",
    "links": [],
    "name": "ExtendedVIFNet",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "OS-FLV-DISABLED",
    "description": "",
    "links": [],
    "name": "FlavorDisabled",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "OS-FLV-EXT-DATA",
    "description": "",
    "links": [],
    "name": "FlavorExtraData",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {

```

```

        "alias": "OS-SCH-HNT",
        "description": "Pass arbitrary key/value pairs to the scheduler.",
        "links": [],
        "name": "SchedulerHints",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "OS-SRV-USG",
        "description": "Adds launched_at and terminated_at on Servers.",
        "links": [],
        "name": "ServerUsage",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-access-ips",
        "description": "Access IPs support.",
        "links": [],
        "name": "AccessIPs",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-admin-actions",
        "description": "Enable admin-only server actions\n\n    Actions
include: resetNetwork, injectNetworkInfo, os-resetState\n    ",
        "links": [],
        "name": "AdminActions",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-admin-password",
        "description": "Admin password management support.",
        "links": [],
        "name": "AdminPassword",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-agents",
        "description": "Agents support.",
        "links": [],
        "name": "Agents",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-aggregates",
        "description": "Admin-only aggregate administration.",
        "links": [],
        "name": "Aggregates",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-assisted-volume-snapshots",
        "description": "Assisted volume snapshots.",
        "links": [],

```

```

        "name": "AssistedVolumeSnapshots",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-attach-interfaces",
        "description": "Attach interface support.",
        "links": [],
        "name": "AttachInterfaces",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-availability-zone",
        "description": "1. Add availability_zone to the Create Server API.
\n      2. Add availability zones describing.\n      ",
        "links": [],
        "name": "AvailabilityZone",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-baremetal-ext-status",
        "description": "",
        "links": [],
        "name": "BareMetalExtStatus",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-baremetal-nodes",
        "description": "Admin-only bare-metal node administration.",
        "links": [],
        "name": "BareMetalNodes",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-block-device-mapping",
        "description": "Block device mapping boot support.",
        "links": [],
        "name": "BlockDeviceMapping",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-block-device-mapping-v2-boot",
        "description": "",
        "links": [],
        "name": "BlockDeviceMappingV2Boot",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-cell-capacities",
        "description": "",
        "links": [],
        "name": "CellCapacities",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    }

```

```

    },
    {
      "alias": "os-cells",
      "description": "Enables cells-related functionality such as adding
neighbor cells,\n    listing neighbor cells, and getting the capabilities of
the local cell.\n    ",
      "links": [],
      "name": "Cells",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-certificates",
      "description": "Certificates support.",
      "links": [],
      "name": "Certificates",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-cloudpipe",
      "description": "Adds actions to create cloudpipe instances.\n\n
When running with the Vlan network mode, you need a mechanism to route\
n    from the public Internet to your vlans. This mechanism is known as a\
n    cloudpipe.\n\n    At the time of creating this class, only OpenVPN is
supported. Support for\n    a SSH Bastion host is forthcoming.\n    ",
      "links": [],
      "name": "Cloudpipe",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-cloudpipe-update",
      "description": "",
      "links": [],
      "name": "CloudpipeUpdate",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-config-drive",
      "description": "Config Drive Extension.",
      "links": [],
      "name": "ConfigDrive",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-console-auth-tokens",
      "description": "Console token authentication support.",
      "links": [],
      "name": "ConsoleAuthTokens",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-console-output",
      "description": "Console log output support, with tailing ability.
",
      "links": [],

```

```
    "name": "ConsoleOutput",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-consoles",
    "description": "Interactive Console support.",
    "links": [],
    "name": "Consoles",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-create-backup",
    "description": "Create a backup of a server.",
    "links": [],
    "name": "CreateBackup",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-create-server-ext",
    "description": "",
    "links": [],
    "name": "Createserverext",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-deferred-delete",
    "description": "Instance deferred delete.",
    "links": [],
    "name": "DeferredDelete",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-evacuate",
    "description": "Enables server evacuation.",
    "links": [],
    "name": "Evacuate",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-extended-evacuate-find-host",
    "description": "",
    "links": [],
    "name": "ExtendedEvacuateFindHost",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-extended-floating-ips",
    "description": "",
    "links": [],
    "name": "ExtendedFloatingIps",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
}
```

```

    {
      "alias": "os-extended-hypervisors",
      "description": "",
      "links": [],
      "name": "ExtendedHypervisors",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-extended-networks",
      "description": "",
      "links": [],
      "name": "ExtendedNetworks",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-extended-quotas",
      "description": "",
      "links": [],
      "name": "ExtendedQuotas",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-extended-rescue-with-image",
      "description": "",
      "links": [],
      "name": "ExtendedRescueWithImage",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-extended-services",
      "description": "",
      "links": [],
      "name": "ExtendedServices",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-extended-services-delete",
      "description": "",
      "links": [],
      "name": "ExtendedServicesDelete",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-extended-status",
      "description": "Extended Status support.",
      "links": [],
      "name": "ExtendedStatus",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-extended-volumes",
      "description": "Extended Volumes support.",
      "links": [],

```



```

    "name": "ExtendedVolumes",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-fixed-ips",
    "description": "Fixed IPs support.",
    "links": [],
    "name": "FixedIPs",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-flavor-access",
    "description": "Flavor access support.",
    "links": [],
    "name": "FlavorAccess",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-flavor-extra-specs",
    "description": "Flavors extra specs support.",
    "links": [],
    "name": "FlavorExtraSpecs",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-flavor-manage",
    "description": "Flavor create/delete API support.",
    "links": [],
    "name": "FlavorManage",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-flavor-rxtx",
    "description": "Support to show the rxtx status of a flavor.",
    "links": [],
    "name": "FlavorRxtx",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-flavor-swap",
    "description": "",
    "links": [],
    "name": "FlavorSwap",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-floating-ip-dns",
    "description": "Floating IP DNS support.",
    "links": [],
    "name": "FloatingIpDns",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },

```

```

    {
      "alias": "os-floating-ip-pools",
      "description": "Floating IPs support.",
      "links": [],
      "name": "FloatingIpPools",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-floating-ips",
      "description": "Floating IPs support.",
      "links": [],
      "name": "FloatingIps",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-floating-ips-bulk",
      "description": "Bulk handling of Floating IPs.",
      "links": [],
      "name": "FloatingIpsBulk",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-fping",
      "description": "Fping Management Extension.",
      "links": [],
      "name": "Fping",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-hide-server-addresses",
      "description": "Support hiding server addresses in certain states.",
      "links": [],
      "name": "HideServerAddresses",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-hosts",
      "description": "Admin-only host administration.",
      "links": [],
      "name": "Hosts",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-hypervisor-status",
      "description": "",
      "links": [],
      "name": "HypervisorStatus",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-hypervisors",
      "description": "Admin-only hypervisor administration.",

```

```

        "links": [],
        "name": "Hypervisors",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-instance-actions",
        "description": "View a log of actions and events taken on an
instance.",
        "links": [],
        "name": "InstanceActions",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-instance_usage_audit_log",
        "description": "Admin-only Task Log Monitoring.",
        "links": [],
        "name": "OSInstanceUsageAuditLog",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-keypairs",
        "description": "Keypair Support.",
        "links": [],
        "name": "Keypairs",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-lock-server",
        "description": "Enable lock/unlock server actions.",
        "links": [],
        "name": "LockServer",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-migrate-server",
        "description": "Enable migrate and live-migrate server actions.",
        "links": [],
        "name": "MigrateServer",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-migrations",
        "description": "Provide data on migrations.",
        "links": [],
        "name": "Migrations",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    },
    {
        "alias": "os-multiple-create",
        "description": "Allow multiple create in the Create Server v2.1
API.",
        "links": [],
        "name": "MultipleCreate",

```

```

    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-networks",
    "description": "Admin-only Network Management Extension.",
    "links": [],
    "name": "Networks",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-networks-associate",
    "description": "Network association support.",
    "links": [],
    "name": "NetworkAssociationSupport",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-pause-server",
    "description": "Enable pause/unpause server actions.",
    "links": [],
    "name": "PauseServer",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-personality",
    "description": "Personality support.",
    "links": [],
    "name": "Personality",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-preserve-ephemeral-rebuild",
    "description": "Allow preservation of the ephemeral partition on
rebuild.",
    "links": [],
    "name": "PreserveEphemeralOnRebuild",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-quota-class-sets",
    "description": "Quota classes management support.",
    "links": [],
    "name": "QuotaClasses",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-quota-sets",
    "description": "Quotas management support.",
    "links": [],
    "name": "Quotas",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  }
}

```

```

    {
      "alias": "os-rescue",
      "description": "Instance rescue mode.",
      "links": [],
      "name": "Rescue",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-security-group-default-rules",
      "description": "Default rules for security group support.",
      "links": [],
      "name": "SecurityGroupDefaultRules",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-security-groups",
      "description": "Security group support.",
      "links": [],
      "name": "SecurityGroups",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-server-diagnostics",
      "description": "Allow Admins to view server diagnostics through
server action.",
      "links": [],
      "name": "ServerDiagnostics",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-server-external-events",
      "description": "Server External Event Triggers.",
      "links": [],
      "name": "ServerExternalEvents",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-server-group-quotas",
      "description": "",
      "links": [],
      "name": "ServerGroupQuotas",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-server-groups",
      "description": "Server group support.",
      "links": [],
      "name": "ServerGroups",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-server-list-multi-status",
      "description": "",

```

```
    "links": [],
    "name": "ServerListMultiStatus",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-server-password",
    "description": "Server password support.",
    "links": [],
    "name": "ServerPassword",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-server-sort-keys",
    "description": "",
    "links": [],
    "name": "ServerSortKeys",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-server-start-stop",
    "description": "",
    "links": [],
    "name": "ServerStartStop",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-services",
    "description": "Services support.",
    "links": [],
    "name": "Services",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-shelve",
    "description": "Instance shelve mode.",
    "links": [],
    "name": "Shelve",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-simple-tenant-usage",
    "description": "Simple tenant usage extension.",
    "links": [],
    "name": "SimpleTenantUsage",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  },
  {
    "alias": "os-suspend-server",
    "description": "Enable suspend/resume server actions.",
    "links": [],
    "name": "SuspendServer",
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "updated": "2014-12-03T00:00:00Z"
  }
```

```

    },
    {
      "alias": "os-tenant-networks",
      "description": "Tenant-based Network Management Extension.",
      "links": [],
      "name": "OSTenantNetworks",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-used-limits",
      "description": "Provide data on limited resources that are being
used.",
      "links": [],
      "name": "UsedLimits",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-used-limits-for-admin",
      "description": "",
      "links": [],
      "name": "UsedLimitsForAdmin",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-user-data",
      "description": "Add user_data to the Create Server API.",
      "links": [],
      "name": "UserData",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-user-quotas",
      "description": "",
      "links": [],
      "name": "UserQuotas",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-virtual-interfaces",
      "description": "Virtual interface support.",
      "links": [],
      "name": "VirtualInterfaces",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-volume-attachment-update",
      "description": "",
      "links": [],
      "name": "VolumeAttachmentUpdate",
      "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
      "updated": "2014-12-03T00:00:00Z"
    },
    {
      "alias": "os-volumes",

```

```
        "description": "Volumes support.",
        "links": [],
        "name": "Volumes",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    }
]
```


4.19.2. Show extension details

Method	URI	Description
GET	/v2.1/{tenant_id}/extensions/{alias}	Shows details for an extension, by alias.

Normal response codes: 200203

4.19.2.1. Request

This table shows the URI parameters for the show extension details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{alias}	String	An alias for the extension name. For example, os-server-external-events.

This operation does not accept a request body.

4.19.2.2. Response

Example 4.112. Show extension details: JSON response

```
{
  "extension": {
    "updated": "2014-12-03T00:00:00Z",
    "name": "ServerExternalEvents",
    "links": [],
    "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
    "alias": "os-server-external-events",
    "description": "Server External Event Triggers."
  }
}
```

4.20. Images

Lists, shows details for, and deletes images. Also sets, lists, shows details for, and deletes image metadata.

An image is a collection of files that you use to create and rebuild a server. By default, operators provide pre-built operating system images. You can also create custom images. See Compute server actions.

By default, the `policy.json` file authorizes all users to view the image size in the `OS-EXT-IMG-SIZE:size` extended attribute.

Method	URI	Description
GET	/v2.1/{tenant_id}/images{?changes-since,server,name,status,minDisk,minRam,type,limit,marker}	Lists IDs, names, and links for available images.
GET	/v2.1/{tenant_id}/images/detail{?changes-since,server,name,status,minDisk,minRam,type,limit,marker}	Lists all details for available images.

Method	URI	Description
GET	/v2.1/{tenant_id}/images/{image_id}	Shows details for an image.
DELETE	/v2.1/{tenant_id}/images/{image_id}	Deletes an image.

4.20.1. List images

Method	URI	Description
GET	/v2.1/{tenant_id}/images{?changes-since,server,name,status,minDisk,minRam,type,limit,marker}	Lists IDs, names, and links for available images.

Normal response codes: 200203

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405)

4.20.1.1. Request

This table shows the URI parameters for the list images request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list images request:

Name	Type	Description
changes-since	DateTime (Optional)	Filters the response by a date and time when the image last changed status. Use this query parameter to check for changes since a previous request rather than re-downloading and re-parsing the full status at each polling interval. If data has changed, the call returns only the items changed since the changes-since time. If data has not changed since the changes-since time, the call returns an empty list. To enable you to keep track of changes, this filter also displays images that were deleted if the changes-since value specifies a date in the last 30 days. Items deleted more than 30 days ago might be returned, but it is not guaranteed. The date and time stamp format is ISO 8601: CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, returns the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. If you omit the time zone, the UTC time zone is assumed.
server	AnyURI (Optional)	Filters the response by a server, as a URL.
name	String (Optional)	Filters the response by an image name, as a string.
status	ImageStatus (Optional)	Filters the response by an image status, as a string. For example, ACTIVE.
minDisk	Int (Optional)	Filters the response by a minimum disk size. For example, 100.
minRam	Int	Filters the response by a minimum RAM size. For example, 512.

Name	Type	Description
	<i>(Optional)</i>	
type	String <i>(Optional)</i>	Filters the response by an image type. For example, snapshot or backup. Possible values: snapshot, backup. Default: ALL.
limit	Int <i>(Optional)</i>	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String <i>(Optional)</i>	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

This operation does not accept a request body.

4.20.1.2. Response

Example 4.113. List images: JSON response

```
{
  "images": [
    {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/v2.1/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        },
        {
          "href": "http://glance.openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "alternate",
          "type": "application/vnd.openstack.image"
        }
      ],
      "name": "fakeimage7"
    },
    {
      "id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
      "links": [
        {
          "href": "http://openstack.example.com/v2.1/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
          "rel": "bookmark"
        }
      ]
    }
  ]
}
```

```

        "href": "http://glance.openstack.example.com/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
        "rel": "alternate",
        "type": "application/vnd.openstack.image"
    },
    ],
    "name": "fakeimage123456"
},
{
    "id": "a2459075-d96c-40d5-893e-577ff92e721c",
    "links": [
        {
            "href": "http://openstack.example.com/v2.1/images/
a2459075-d96c-40d5-893e-577ff92e721c",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/images/a2459075-
d96c-40d5-893e-577ff92e721c",
            "rel": "bookmark"
        },
        {
            "href": "http://glance.openstack.example.com/images/
a2459075-d96c-40d5-893e-577ff92e721c",
            "rel": "alternate",
            "type": "application/vnd.openstack.image"
        }
    ],
    "name": "fakeimage123456"
},
{
    "id": "a440c04b-79fa-479c-bed1-0b816eaec379",
    "links": [
        {
            "href": "http://openstack.example.com/v2.1/images/
a440c04b-79fa-479c-bed1-0b816eaec379",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/images/
a440c04b-79fa-479c-bed1-0b816eaec379",
            "rel": "bookmark"
        },
        {
            "href": "http://glance.openstack.example.com/images/
a440c04b-79fa-479c-bed1-0b816eaec379",
            "rel": "alternate",
            "type": "application/vnd.openstack.image"
        }
    ],
    "name": "fakeimage6"
},
{
    "id": "c905cedb-7281-47e4-8a62-f26bc5fc4c77",
    "links": [
        {
            "href": "http://openstack.example.com/v2.1/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77",
            "rel": "self"
        },
    ],

```

```

        {
            "href": "http://openstack.example.com/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77",
            "rel": "bookmark"
        },
        {
            "href": "http://glance.openstack.example.com/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77",
            "rel": "alternate",
            "type": "application/vnd.openstack.image"
        }
    ],
    "name": "fakeimage123456"
},
{
    "id": "cedef40a-ed67-4d10-800e-17455edce175",
    "links": [
        {
            "href": "http://openstack.example.com/v2.1/images/
cedef40a-ed67-4d10-800e-17455edce175",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/images/cedef40a-
ed67-4d10-800e-17455edce175",
            "rel": "bookmark"
        },
        {
            "href": "http://glance.openstack.example.com/images/
cedef40a-ed67-4d10-800e-17455edce175",
            "rel": "alternate",
            "type": "application/vnd.openstack.image"
        }
    ],
    "name": "fakeimage123456"
},
{
    "id": "76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
    "links": [
        {
            "href": "http://openstack.example.com/v2.1/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/images/76fa36fc-
c930-4bf3-8c8a-ea2a2420deb6",
            "rel": "bookmark"
        },
        {
            "href": "http://glance.openstack.example.com/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
            "rel": "alternate",
            "type": "application/vnd.openstack.image"
        }
    ],
    "name": "fakeimage123456"
}
]
}

```

This operation does not return a response body.

4.20.2. List images details

Method	URI	Description
GET	/v2.1/{tenant_id}/images/detail{?changes-since,server,name,status,minDisk,minRam,type,limit,marker}	Lists all details for available images.

Normal response codes: 200203

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405)

4.20.2.1. Request

This table shows the URI parameters for the list images details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list images details request:

Name	Type	Description
changes-since	DateTime (Optional)	Filters the response by a date and time when the image last changed status. Use this query parameter to check for changes since a previous request rather than re-downloading and re-parsing the full status at each polling interval. If data has changed, the call returns only the items changed since the changes-since time. If data has not changed since the changes-since time, the call returns an empty list. To enable you to keep track of changes, this filter also displays images that were deleted if the changes-since value specifies a date in the last 30 days. Items deleted more than 30 days ago might be returned, but it is not guaranteed. The date and time stamp format is ISO 8601: CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, returns the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. If you omit the time zone, the UTC time zone is assumed.
server	AnyURI (Optional)	Filters the response by a server, as a URL.
name	String (Optional)	Filters the response by an image name, as a string.
status	ImageStatus (Optional)	Filters the response by an image status, as a string. For example, ACTIVE.
minDisk	Int (Optional)	Filters the response by a minimum disk size. For example, 100.
minRam	Int	Filters the response by a minimum RAM size. For example, 512.

Name	Type	Description
	<i>(Optional)</i>	
type	String <i>(Optional)</i>	Filters the response by an image type. For example, snapshot or backup. Possible values: snapshot, backup. Default: ALL.
limit	Int <i>(Optional)</i>	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String <i>(Optional)</i>	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

This operation does not accept a request body.

4.20.2.2. Response

Example 4.114. List images details: JSON response

```
{
  "images": [
    {
      "OS-EXT-IMG-SIZE:size": "74185822",
      "created": "2011-01-01T01:02:03Z",
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/v2.1/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        },
        {
          "href": "http://glance.openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "alternate",
          "type": "application/vnd.openstack.image"
        }
      ],
      "metadata": {
        "architecture": "x86_64",
        "auto_disk_config": true,
        "kernel_id": "nokernel",
        "ramdisk_id": "nokernel"
      },
      "minDisk": 0,
      "minRam": 0,
      "name": "fakeimage7",
      "progress": 100,
      "status": "ACTIVE",
      "updated": "2011-01-01T01:02:03Z"
    },
    {

```

```

        "OS-EXT-IMG-SIZE:size": "74185821",
        "created": "2011-01-01T01:02:03Z",
        "id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
        "links": [
            {
                "href": "http://openstack.example.com/v2.1/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
                "rel": "alternate",
                "type": "application/vnd.openstack.image"
            }
        ],
        "metadata": {
            "architecture": "x86_64",
            "kernel_id": "nokernel",
            "ramdisk_id": "nokernel"
        },
        "minDisk": 0,
        "minRam": 0,
        "name": "fakeimage123456",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    },
    {
        "created": "2011-01-01T01:02:03Z",
        "id": "a2459075-d96c-40d5-893e-577ff92e721c",
        "links": [
            {
                "href": "http://openstack.example.com/v2.1/images/
a2459075-d96c-40d5-893e-577ff92e721c",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/images/a2459075-
d96c-40d5-893e-577ff92e721c",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/images/
a2459075-d96c-40d5-893e-577ff92e721c",
                "rel": "alternate",
                "type": "application/vnd.openstack.image"
            }
        ],
        "metadata": {
            "kernel_id": "nokernel",
            "ramdisk_id": "nokernel"
        },
        "minDisk": 0,
        "minRam": 0,

```

```

        "name": "fakeimage123456",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    },
    {
        "created": "2011-01-01T01:02:03Z",
        "id": "a440c04b-79fa-479c-bed1-0b816eae379",
        "links": [
            {
                "href": "http://openstack.example.com/v2.1/images/a440c04b-79fa-479c-bed1-0b816eae379",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/images/a440c04b-79fa-479c-bed1-0b816eae379",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/images/a440c04b-79fa-479c-bed1-0b816eae379",
                "rel": "alternate",
                "type": "application/vnd.openstack.image"
            }
        ],
        "metadata": {
            "architecture": "x86_64",
            "auto_disk_config": false,
            "kernel_id": "nokernel",
            "ramdisk_id": "nokernel"
        },
        "minDisk": 0,
        "minRam": 0,
        "name": "fakeimage6",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    },
    {
        "created": "2011-01-01T01:02:03Z",
        "id": "c905cedb-7281-47e4-8a62-f26bc5fc4c77",
        "links": [
            {
                "href": "http://openstack.example.com/v2.1/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "alternate",
                "type": "application/vnd.openstack.image"
            }
        ],
    },
],

```

```

        "metadata": {
            "kernel_id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
            "ramdisk_id": null
        },
        "minDisk": 0,
        "minRam": 0,
        "name": "fakeimage123456",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    },
    {
        "created": "2011-01-01T01:02:03Z",
        "id": "cedef40a-ed67-4d10-800e-17455edce175",
        "links": [
            {
                "href": "http://openstack.example.com/v2.1/images/
cedef40a-ed67-4d10-800e-17455edce175",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/images/cedef40a-
ed67-4d10-800e-17455edce175",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/images/
cedef40a-ed67-4d10-800e-17455edce175",
                "rel": "alternate",
                "type": "application/vnd.openstack.image"
            }
        ],
        "metadata": {
            "kernel_id": "nokernel",
            "ramdisk_id": "nokernel"
        },
        "minDisk": 0,
        "minRam": 0,
        "name": "fakeimage123456",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    },
    {
        "created": "2011-01-01T01:02:03Z",
        "id": "76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
        "links": [
            {
                "href": "http://openstack.example.com/v2.1/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/images/76fa36fc-
c930-4bf3-8c8a-ea2a2420deb6",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",

```

```
        "rel": "alternate",
        "type": "application/vnd.openstack.image"
    },
    "metadata": {
        "kernel_id": "nokernel",
        "ramdisk_id": "nokernel"
    },
    "minDisk": 0,
    "minRam": 0,
    "name": "fakeimage123456",
    "progress": 100,
    "status": "ACTIVE",
    "updated": "2011-01-01T01:02:03Z"
}
]
```

This operation does not return a response body.

4.20.3. Show image details

Method	URI	Description
GET	/v2.1/{tenant_id}/images/{image_id}	Shows details for an image.

Normal response codes: 200203

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.20.3.1. Request

This table shows the URI parameters for the show image details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

4.20.3.2. Response

Example 4.115. Show image details: JSON response

```
{
  "image": {
    "OS-EXT-IMG-SIZE:size": "74185822",
    "created": "2011-01-01T01:02:03Z",
    "id": "70a599e0-31e7-49b7-b260-868f441e862b",
    "links": [
      {
        "href": "http://openstack.example.com/v2.1/images/70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "bookmark"
      },
      {
        "href": "http://glance.openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "alternate",
        "type": "application/vnd.openstack.image"
      }
    ],
    "metadata": {
      "architecture": "x86_64",
      "auto_disk_config": true,
      "kernel_id": "nokernel",
      "ramdisk_id": "nokernel"
    },
    "minDisk": 0,
  }
}
```

```
    "minRam": 0,  
    "name": "fakeimage7",  
    "progress": 100,  
    "status": "ACTIVE",  
    "updated": "2011-01-01T01:02:03Z"  
  }  
}
```

4.20.4. Delete image

Method	URI	Description
DELETE	/v2.1/{tenant_id}/images/{image_id}	Deletes an image.

Normal response codes: 204

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.20.4.1. Request

This table shows the URI parameters for the delete image request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

4.21. Image metadata

Shows details for, sets, updates, and deletes image metadata or metadata items.

Method	URI	Description
GET	/v2.1/{tenant_id}/images/{image_id}/metadata	Shows metadata for an image.
PUT	/v2.1/{tenant_id}/images/{image_id}/metadata	Creates or replaces metadata for an image.
POST	/v2.1/{tenant_id}/images/{image_id}/metadata	Updates metadata items, by key, for an image.
GET	/v2.1/{tenant_id}/images/{image_id}/metadata/{key}	Shows details for a metadata item, by key, for an image.
PUT	/v2.1/{tenant_id}/images/{image_id}/metadata/{key}	Creates or updates a metadata item, by key, for an image.
DELETE	/v2.1/{tenant_id}/images/{image_id}/metadata/{key}	Deletes a metadata item, by key, for an image.

4.21.1. Show image metadata

Method	URI	Description
GET	/v2.1/{tenant_id}/images/{image_id}/metadata	Shows metadata for an image.

Normal response codes: 200203

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.21.1.1. Request

This table shows the URI parameters for the show image metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

4.21.1.2. Response

Example 4.116. Show image metadata: JSON response

```
{
  "metadata": {
    "architecture": "x86_64",
    "auto_disk_config": true,
    "kernel_id": "nokernel",
    "ramdisk_id": "nokernel"
  }
}
```

This operation does not return a response body.

4.21.2. Create or replace image metadata

Method	URI	Description
PUT	/v2.1/{tenant_id}/images/{image_id}/metadata	Creates or replaces metadata for an image.

Replaces items that match keys. If you omit a key that already exists, this key retains its value.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.21.2.1. Request

This table shows the URI parameters for the create or replace image metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{image_id}	UUID	The UUID of the image.

Example 4.117. Create or replace image metadata: JSON request

```
{
  "metadata": {
    "auto_disk_config": true,
    "Label": "Changed"
  }
}
```

4.21.2.2. Response

Example 4.118. Create or replace image metadata: JSON response

```
{
  "metadata": {
    "Label": "Changed",
    "auto_disk_config": true
  }
}
```

4.21.3. Update image metadata items

Method	URI	Description
POST	/v2.1/{tenant_id}/images/{image_id}/metadata	Updates metadata items, by key, for an image.

Replaces items that match keys. Does not modify items not in the request.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.21.3.1. Request

This table shows the URI parameters for the update image metadata items request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{image_id}	UUID	The UUID of the image.

Example 4.119. Update image metadata items: JSON request

```
{
  "metadata": {
    "kernel_id": false,
    "Label": "UpdatedImage"
  }
}
```

4.21.3.2. Response

Example 4.120. Update image metadata items: JSON response

```
{
  "metadata": {
    "Label": "UpdatedImage",
    "architecture": "x86_64",
    "auto_disk_config": true,
    "kernel_id": false,
    "ramdisk_id": "nokernel"
  }
}
```

This operation does not return a response body.

4.21.4. Show image metadata item details

Method	URI	Description
GET	/v2.1/{tenant_id}/images/{image_id}/metadata/{key}	Shows details for a metadata item, by key, for an image.

Normal response codes: 200203

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.21.4.1. Request

This table shows the URI parameters for the show image metadata item details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{image_id}	UUID	The UUID of the image.
{key}	String	The metadata item key, as a string. Maximum length is 255 characters.

This operation does not accept a request body.

4.21.4.2. Response

Example 4.121. Show image metadata item details: JSON response

```
{
  "meta": {
    "architecture": "x86_64",
    "auto_disk_config": true,
    "kernel_id": "nokernel",
    "ramdisk_id": "nokernel"
  }
}
```

4.21.5. Create or update image metadata item

Method	URI	Description
PUT	/v2.1/{tenant_id}/images/{image_id}/metadata/{key}	Creates or updates a metadata item, by key, for an image.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.21.5.1. Request

This table shows the URI parameters for the create or update image metadata item request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{image_id}	UUID	The UUID of the image.
{key}	String	The metadata item key, as a string. Maximum length is 255 characters.

Example 4.122. Create or update image metadata item: JSON request

```
{
  "meta": {
    "auto_disk_config": false
  }
}
```

4.21.5.2. Response

Example 4.123. Create or update image metadata item: JSON response

```
{
  "meta": {
    "auto_disk_config": false
  }
}
```

4.21.6. Delete image metadata item

Method	URI	Description
DELETE	/v2.1/{tenant_id}/images/{image_id}/metadata/{key}	Deletes a metadata item, by key, for an image.

Normal response codes: 204

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), buildInProgress (409)

4.21.6.1. Request

This table shows the URI parameters for the delete image metadata item request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{image_id}	UUID	The UUID of the image.
{key}	String	The metadata item key, as a string. Maximum length is 255 characters.

This operation does not accept a request body.

4.22. Guest agents (os-agents)

Creates, lists, updates, and deletes guest agent builds. Use guest agents to access files on the disk, configure networking, or run other applications or scripts in the guest while the agent runs. This hypervisor-specific extension is not currently enabled for KVM. Use of guest agents is possible only if the underlying service provider uses the Xen driver.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-agents	Creates an agent build.
GET	/v2.1/{tenant_id}/os-agents	Lists agent builds.
DELETE	/v2.1/{tenant_id}/os-agents	Deletes an existing agent build.
PUT	/v2.1/{tenant_id}/os-agents/{agent_build_id}	Updates an agent build.

4.22.1. Create agent build

Method	URI	Description
POST	/v2.1/{tenant_id}/os-agents	Creates an agent build.

Normal response codes: 201

4.22.1.1. Request

This table shows the URI parameters for the create agent build request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.124. Create agent build: JSON request

```
{
  "agent": {
    "hypervisor": "hypervisor",
    "os": "os",
    "architecture": "x86",
    "version": "8.0",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "url": "http://example.com/path/to/resource"
  }
}
```

4.22.1.2. Response

Example 4.125. Create agent build: JSON response

```
{
  "agent": {
    "agent_id": 1,
    "architecture": "x86",
    "hypervisor": "hypervisor",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "os": "os",
    "url": "http://example.com/path/to/resource",
    "version": "8.0"
  }
}
```

4.22.2. List agent builds

Method	URI	Description
GET	/v2.1/{tenant_id}/os-agents	Lists agent builds.

Normal response codes: 200

4.22.2.1. Request

This table shows the URI parameters for the list agent builds request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

4.22.2.2. Response

Example 4.126. List agent builds: JSON response

```
{
  "agents": [
    {
      "agent_id": 1,
      "architecture": "x86",
      "hypervisor": "hypervisor",
      "md5hash": "add6bb58e139be103324d04d82d8f545",
      "os": "os",
      "url": "http://example.com/path/to/resource",
      "version": "8.0"
    }
  ]
}
```


4.22.3. Delete agent build

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-agents	Deletes an existing agent build.

Normal response codes: 202

4.22.3.1. Request

This table shows the URI parameters for the delete agent build request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.22.4. Update agent build

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-agents/{agent_build_id}	Updates an agent build.

Normal response codes: 200

4.22.4.1. Request

This table shows the URI parameters for the update agent build request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{agent_build_id}	UUID	The UUID of the agent build.

Example 4.127. Update agent build: JSON request

```
{
  "para": {
    "url": "http://example.com/path/to/resource",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "version": "7.0"
  }
}
```

4.22.4.2. Response

Example 4.128. Update agent build: JSON response

```
{
  "agent": {
    "agent_id": "1",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "url": "http://example.com/path/to/resource",
    "version": "7.0"
  }
}
```

4.23. Host aggregates (os-aggregates, action)

Creates and manages host aggregates. An aggregate assigns metadata to groups of compute nodes. Aggregates are only visible to the cloud provider.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-aggregates	Creates an aggregate in an availability zone.
GET	/v2.1/{tenant_id}/os-aggregates	Lists all aggregates. Includes the ID, name, and availability zone for each aggregate.
GET	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Shows details for an aggregate. Details include hosts and metadata.
DELETE	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Deletes an aggregate.

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Updates either or both the name and availability zone for an aggregate.
POST	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action	Adds a host to an aggregate.
POST	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action	Creates or replaces metadata for an aggregate.
POST	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action	Removes a host from an aggregate.

4.23.1. Create aggregate

Method	URI	Description
POST	/v2.1/{tenant_id}/os-aggregates	Creates an aggregate in an availability zone.

Normal response codes: 200

4.23.1.1. Request

This table shows the URI parameters for the create aggregate request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.129. Create aggregate: JSON request

```
{
  "aggregate": {
    "name": "name",
    "availability_zone": "nova"
  }
}
```

4.23.1.2. Response

Example 4.130. Create aggregate: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2013-08-18T12:17:55.751757",
    "deleted": false,
    "deleted_at": null,
    "id": 1,
    "name": "name",
    "updated_at": null
  }
}
```

4.23.2. List aggregates

Method	URI	Description
GET	/v2.1/{tenant_id}/os-aggregates	Lists all aggregates. Includes the ID, name, and availability zone for each aggregate.

Normal response codes: 200

4.23.2.1. Request

This table shows the URI parameters for the list aggregates request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.131. List aggregates: JSON request

```
{
  "aggregate": {
    "name": "name",
    "availability_zone": "nova"
  }
}
```

4.23.2.2. Response

Example 4.132. List aggregates: JSON response

```
{
  "aggregates": [
    {
      "availability_zone": "nova",
      "created_at": "2013-08-18T12:17:56.856455",
      "deleted": false,
      "deleted_at": null,
      "hosts": [],
      "id": 1,
      "metadata": {
        "availability_zone": "nova"
      },
      "name": "name",
      "updated_at": null
    }
  ]
}
```

4.23.3. Show aggregate details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Shows details for an aggregate. Details include hosts and metadata.

Normal response codes: 200

4.23.3.1. Request

This table shows the URI parameters for the show aggregate details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{aggregate_id}	Int	The aggregate ID.

Example 4.133. Show aggregate details: JSON request

```
{
  "aggregate": {
    "name": "name",
    "availability_zone": "nova"
  }
}
```

4.23.3.2. Response

Example 4.134. Show aggregate details: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2013-08-18T12:17:56.380226",
    "deleted": false,
    "deleted_at": null,
    "hosts": [],
    "id": 1,
    "metadata": {
      "availability_zone": "nova"
    },
    "name": "name",
    "updated_at": null
  }
}
```

4.23.4. Delete aggregate

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Deletes an aggregate.

Normal response codes: 200

4.23.4.1. Request

This table shows the URI parameters for the delete aggregate request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{aggregate_id}	Int	The aggregate ID.

This operation does not accept a request body.

4.23.5. Update aggregate

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Updates either or both the name and availability zone for an aggregate.

Normal response codes: 200

4.23.5.1. Request

This table shows the URI parameters for the update aggregate request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{aggregate_id}	Int	The aggregate ID.

Example 4.135. Update aggregate: JSON request

```
{
  "aggregate": {
    "name": "newname",
    "availability_zone": "nova2"
  }
}
```

4.23.5.2. Response

Example 4.136. Update aggregate: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova2",
    "created_at": "2013-08-18T12:17:56.259751",
    "deleted": false,
    "deleted_at": null,
    "hosts": [],
    "id": 1,
    "metadata": {
      "availability_zone": "nova2"
    },
    "name": "newname",
    "updated_at": "2013-08-18T12:17:56.286720"
  }
}
```


4.23.6. Add host

Method	URI	Description
POST	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action	Adds a host to an aggregate.

Specify the `add_host` action in the request body.

Normal response codes: 200

4.23.6.1. Request

This table shows the URI parameters for the add host request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{aggregate_id}	Int	The aggregate ID.

Example 4.137. Add host: JSON request

```
{
  "add_host": {
    "host": "21549b2f665945baaa7101926a00143c"
  }
}
```

4.23.6.2. Response

Example 4.138. Add host: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2013-08-18T12:17:56.297823",
    "deleted": false,
    "deleted_at": null,
    "hosts": [
      "21549b2f665945baaa7101926a00143c"
    ],
    "id": 1,
    "metadata": {
      "availability_zone": "nova"
    },
    "name": "name",
    "updated_at": null
  }
}
```

4.23.7. Create or update aggregate metadata

Method	URI	Description
POST	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action	Creates or replaces metadata for an aggregate.

Specify the `set_metadata` action in the request body.

Normal response codes: 200

4.23.7.1. Request

This table shows the URI parameters for the create or update aggregate metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{aggregate_id}	Int	The aggregate ID.

Example 4.139. Create or update aggregate metadata: JSON request

```
{
  "set_metadata": {
    "metadata": {
      "key": "value"
    }
  }
}
```

4.23.7.2. Response

Example 4.140. Create or update aggregate metadata: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2013-08-18T12:17:55.959571",
    "deleted": false,
    "deleted_at": null,
    "hosts": [],
    "id": 1,
    "metadata": {
      "availability_zone": "nova",
      "key": "value"
    },
    "name": "name",
    "updated_at": null
  }
}
```

4.23.8. Remove host

Method	URI	Description
POST	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action	Removes a host from an aggregate.

Specify the `remove_host` action in the request body.

Normal response codes: 200

4.23.8.1. Request

This table shows the URI parameters for the remove host request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{aggregate_id}	Int	The aggregate ID.

Example 4.141. Remove host: JSON request

```
{
  "remove_host": {
    "host": "bf1454b3d71145d49fca2101c56c728d"
  }
}
```

4.23.8.2. Response

Example 4.142. Remove host: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2013-08-18T12:17:56.990581",
    "deleted": false,
    "deleted_at": null,
    "hosts": [],
    "id": 1,
    "metadata": {
      "availability_zone": "nova"
    },
    "name": "name",
    "updated_at": null
  }
}
```

4.24. Assisted volume snapshots (os-assisted-volume-snapshots)

Creates and deletes snapshots through an emulator/hypervisor. The qcow2 file format is supported.

An internal snapshot that lacks storage such as NFS or GlusterFS can use an emulator/hypervisor to add the snapshot feature.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-assisted-volume-snapshots	Creates an assisted volume snapshot.
DELETE	/v2.1/{tenant_id}/os-assisted-volume-snapshots/{snapshot_id}{?delete_info}	Deletes an assisted volume snapshot.

4.24.1. Create assisted volume snapshots

Method	URI	Description
POST	/v2.1/{tenant_id}/os-assisted-volume-snapshots	Creates an assisted volume snapshot.

Normal response codes: 200

4.24.1.1. Request

This table shows the URI parameters for the create assisted volume snapshots request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.143. Create assisted volume snapshots: JSON request

```
{
  "snapshot": {
    "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "create_info": {
      "snapshot_id": "421752a6-acf6-4b2d-bc7a-119f9148cd8c",
      "type": "qcow2",
      "new_file": "new_file_name"
    }
  }
}
```

4.24.1.2. Response

Example 4.144. Create assisted volume snapshots: JSON response

```
{
  "snapshot": {
    "id": 100,
    "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c"
  }
}
```

4.24.2. Delete assisted volume snapshot

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-assisted-volume-snapshots/{snapshot_id}{?delete_info}	Deletes an assisted volume snapshot.

To make this request, add the `delete_info` query parameter to the URI, as follows:

```
DELETE /os-assisted-volume-snapshots?delete_info='{ "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c" }'
```

Normal response codes: 204

4.24.2.1. Request

This table shows the URI parameters for the delete assisted volume snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

This table shows the query parameters for the delete assisted volume snapshot request:

Name	Type	Description
delete_info	String (Required)	Information for snapshot deletion. Include the ID of the associated volume. For example: <pre>DELETE /os-assisted-volume-snapshots?delete_info='{ "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c" }'</pre>

This operation does not accept a request body.

4.25. Availability zones (os-availability-zone)

Shows availability zone information.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-availability-zone	Gets availability zone information.
GET	/v2.1/{tenant_id}/os-availability-zone/detail	Gets detailed availability zone information.

4.25.1. Get availability zone information

Method	URI	Description
GET	/v2.1/{tenant_id}/os-availability-zone	Gets availability zone information.

Normal response codes: 200

4.25.1.1. Request

This table shows the URI parameters for the get availability zone information request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.25.1.2. Response

Example 4.145. Get availability zone information: JSON response

```
{
  "availabilityZoneInfo": [
    {
      "zoneState": {
        "available": true
      },
      "hosts": null,
      "zoneName": "nova"
    }
  ]
}
```

4.25.2. Get detailed availability zone information

Method	URI	Description
GET	/v2.1/{tenant_id}/os-availability-zone/detail	Gets detailed availability zone information.

Normal response codes: 200

4.25.2.1. Request

This table shows the URI parameters for the get detailed availability zone information request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.25.2.2. Response

Example 4.146. Get detailed availability zone information: JSON response

```
{
  "availabilityZoneInfo": [
    {
      "zoneState": {
        "available": true
      },
      "hosts": {
        "test-host": {
          "nova-conductor": {
            "available": true,
            "active": true,
            "updated_at": "2015-04-16T08:58:55.000000"
          },
          "nova-cert": {
            "available": true,
            "active": true,
            "updated_at": "2015-04-16T08:58:55.000000"
          },
          "nova-consoleauth": {
            "available": true,
            "active": true,
            "updated_at": "2015-04-16T08:58:55.000000"
          },
          "nova-scheduler": {
            "available": true,
            "active": true,
            "updated_at": "2015-04-16T08:58:55.000000"
          },
          "nova-network": {
            "available": true,
            "active": true,
            "updated_at": "2015-04-16T08:58:54.000000"
          }
        }
      }
    }
  ]
}
```



```

    }
  },
  "zoneName": "internal"
},
{
  "zoneState": {
    "available": true
  },
  "hosts": {
    "test-host": {
      "nova-compute": {
        "available": true,
        "active": true,
        "updated_at": "2015-04-16T08:58:56.000000"
      }
    }
  },
  "zoneName": "nova"
}
]
}

```

4.26. Bare metal nodes (os-baremetal-nodes)

Bare metal nodes.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes	Adds a bare metal node to a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes	Lists the bare metal nodes that are associated with a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action	Adds an interface to a bare metal node that is associated with a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action	Deletes an interface from a bare metal node that is associated with a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}	Shows details for a bare metal node.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}	Deletes a bare metal node from a server.

4.26.1. Add bare metal node

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes	Adds a bare metal node to a server.

Preconditions

- You can add a bare metal node to a server with an ACTIVE, PAUSED, SHUTOFF, VERIFY_RESIZE, or SOFT_DELETED status.
- You can add a bare metal node to a server with a status that is not locked.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.26.1.1. Request

This table shows the URI parameters for the add bare metal node request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.147. Add bare metal node: JSON request

```
{
  "node": {
    "service_host": "host",
    "cpus": 8,
    "memory_mb": 8192,
    "local_gb": 128,
    "pm_address": "10.1.2.3",
    "pm_user": "pm_user",
    "pm_password": "pm_pass",
    "terminal_port": 8000
  }
}
```

Example 4.148. Add bare metal node: JSON request

```
{
  "node": {
    "service_host": "host",
    "cpus": 8,
    "memory_mb": 8192,
    "local_gb": 128,
    "pm_address": "10.1.2.3",
    "pm_user": "pm_user",
    "pm_password": "pm_pass",
    "prov_mac_address": "12:34:56:78:90:ab",
  }
}
```

```
    "terminal_port": 8000
  }
}
```

4.26.1.2. Response

Example 4.149. Add bare metal node: JSON response

```
{
  "node": {
    "cpus": 8,
    "id": 1,
    "instance_uuid": null,
    "interfaces": [],
    "local_gb": 128,
    "memory_mb": 8192,
    "pm_address": "10.1.2.3",
    "pm_user": "pm_user",
    "pxe_config_path": null,
    "service_host": "host",
    "task_state": null,
    "terminal_port": 8000,
    "updated_at": null,
    "uuid": "73d35253-b6fb-4c83-b8eb-0229336e79b6"
  }
}
```

Example 4.150. Add bare metal node: JSON response

```
{
  "node": {
    "cpus": 8,
    "id": 1,
    "instance_uuid": null,
    "interfaces": [
      {
        "address": "12:34:56:78:90:ab",
        "datapath_id": null,
        "id": 1,
        "port_no": null
      }
    ],
    "local_gb": 128,
    "memory_mb": 8192,
    "pm_address": "10.1.2.3",
    "pm_user": "pm_user",
    "pxe_config_path": null,
    "service_host": "host",
    "task_state": null,
    "terminal_port": 8000,
    "updated_at": null,
    "uuid": "0a130464-bccc-4e36-b9d3-9a8c98e636ae"
  }
}
```

4.26.2. List bare metal nodes

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes	Lists the bare metal nodes that are associated with a server.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.26.2.1. Request

This table shows the URI parameters for the list bare metal nodes request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.26.2.2. Response

Example 4.151. List bare metal nodes: JSON response

```
{
  "nodes": [
    {
      "cpus": 8,
      "id": 1,
      "instance_uuid": null,
      "interfaces": [
        {
          "address": "aa:aa:aa:aa:aa:aa",
          "datapath_id": null,
          "id": 1,
          "port_no": null
        }
      ],
      "local_gb": 128,
      "memory_mb": 8192,
      "pm_address": "10.1.2.3",
      "pm_user": "pm_user",
      "pxe_config_path": null,
      "service_host": "host",
      "task_state": null,
      "terminal_port": 8000,
      "updated_at": null,
      "uuid": "6fae68da-108b-4a9d-87c4-88831ee1241b"
    }
  ]
}
```

4.26.3. Add interface to bare metal node

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action	Adds an interface to a bare metal node that is associated with a server.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.26.3.1. Request

This table shows the URI parameters for the add interface to bare metal node request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.152. Add interface to bare metal node: JSON request

```
{
  "add_interface": {
    "address": "aa:aa:aa:aa:aa:aa"
  }
}
```

4.26.3.2. Response

Example 4.153. Add interface to bare metal node: JSON response

```
{
  "interface": {
    "address": "aa:aa:aa:aa:aa:aa",
    "datapath_id": null,
    "id": 1,
    "port_no": null
  }
}
```

4.26.4. Delete interface from bare metal node

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action	Deletes an interface from a bare metal node that is associated with a server.

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.26.4.1. Request

This table shows the URI parameters for the delete interface from bare metal node request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

Example 4.154. Delete interface from bare metal node: JSON request

```
{
  "remove_interface": {
    "address": "aa:aa:aa:aa:aa:aa"
  }
}
```

4.26.5. Show bare metal node details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}	Shows details for a bare metal node.

Preconditions

- The bare metal node must be associated with the server.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.26.5.1. Request

This table shows the URI parameters for the show bare metal node details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{node_id}	String	The node ID.

This operation does not accept a request body.

4.26.5.2. Response

Example 4.155. Show bare metal node details: JSON response

```
{
  "node": {
    "cpus": 8,
    "id": 1,
    "instance_uuid": null,
    "interfaces": [
      {
        "address": "aa:aa:aa:aa:aa:aa",
        "datapath_id": null,
        "id": 1,
        "port_no": null
      }
    ],
    "local_gb": 128,
    "memory_mb": 8192,
    "pm_address": "10.1.2.3",
    "pm_user": "pm_user",
    "pxe_config_path": null,
    "service_host": "host",
    "task_state": null,
    "terminal_port": 8000,
    "updated_at": null,
  }
}
```

```
    "uuid": "c862b836-c7c1-4f7f-8081-6766fa9cf38b"  
  }  
}
```


4.26.6. Delete bare metal node

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}	Deletes a bare metal node from a server.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

4.26.6.1. Request

This table shows the URI parameters for the delete bare metal node request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{node_id}	String	The node ID.

This operation does not accept a request body.

4.27. Cells (os-cells, capacities)

Adds neighbor cells, lists neighbor cells, and shows the capabilities of the local cell.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-cells	Lists cells.
GET	/v2.1/{tenant_id}/os-cells	Lists cells with details.
GET	/v2.1/{tenant_id}/os-cells/{cell_id}	Shows data for a cell.
GET	/v2.1/{tenant_id}/os-cells/{cell_id}/capacities	Shows capacities for a cell.

4.27.1. List cells

Method	URI	Description
GET	/v2.1/{tenant_id}/os-cells	Lists cells.

When cells are not enabled, the call returns the Not Implemented (501) response code.

Normal response codes: 200

Error response codes: notImplemented (501)

4.27.1.1. Request

This table shows the URI parameters for the list cells request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.27.1.2. Response

Example 4.156. List cells: JSON response

```
{
  "cells": [
    {
      "name": "cell1",
      "rpc_host": null,
      "rpc_port": null,
      "type": "child",
      "username": "username1"
    },
    {
      "name": "cell3",
      "rpc_host": null,
      "rpc_port": null,
      "type": "child",
      "username": "username3"
    },
    {
      "name": "cell5",
      "rpc_host": null,
      "rpc_port": null,
      "type": "child",
      "username": "username5"
    },
    {
      "name": "cell2",
      "rpc_host": null,
      "rpc_port": null,
      "type": "parent",
      "username": "username2"
    }
  ],
}
```

```
{
  {
    "name": "cell4",
    "rpc_host": null,
    "rpc_port": null,
    "type": "parent",
    "username": "username4"
  }
}
```

4.27.2. List cells with details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-cells	Lists cells with details.

When cells are not enabled, the call returns the Not Implemented (501) response code.

Normal response codes: 200

Error response codes: notImplemented (501)

4.27.2.1. Request

This table shows the URI parameters for the list cells with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.27.2.2. Response

Example 4.157. List cells with details: JSON response

```
{
  "cells": []
}
```

4.27.3. Show cell data

Method	URI	Description
GET	/v2.1/{tenant_id}/os-cells/{cell_id}	Shows data for a cell.

When cells are not enabled, the call returns the Not Implemented (501) response code.

Normal response codes: 200

Error response codes: notImplemented (501)

4.27.3.1. Request

This table shows the URI parameters for the show cell data request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{cell_id}	UUID	The UUID of the cell.

This operation does not accept a request body.

4.27.3.2. Response

Example 4.158. Show cell data: JSON response

```
{
  "cell": {
    "name": "cell3",
    "rpc_host": null,
    "rpc_port": null,
    "type": "child",
    "username": "username3"
  }
}
```

4.27.4. Show cell capacities

Method	URI	Description
GET	/v2.1/{tenant_id}/os-cells/{cell_id}/capacities	Shows capacities for a cell.

When cells are not enabled, the call returns the Not Implemented (501) response code.

Normal response codes: 200

Error response codes: notImplemented (501)

4.27.4.1. Request

This table shows the URI parameters for the show cell capacities request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{cell_id}	UUID	The UUID of the cell.

This operation does not accept a request body.

4.27.4.2. Response

Example 4.159. Show cell capacities: JSON response

```
{
  "cell": {
    "capacities": {
      "disk_free": {
        "total_mb": 1052672,
        "units_by_mb": {
          "0": 0,
          "163840": 5,
          "20480": 46,
          "40960": 23,
          "81920": 11
        }
      },
      "ram_free": {
        "total_mb": 7680,
        "units_by_mb": {
          "16384": 0,
          "2048": 3,
          "4096": 1,
          "512": 13,
          "8192": 0
        }
      }
    }
  }
}
```

4.28. Root certificates (os-certificates)

Creates and shows details for a root certificate.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-certificates	Creates a certificate.
GET	/v2.1/{tenant_id}/os-certificates/{certificate_id}	Shows details for a certificate.

4.28.1. Create certificate

Method	URI	Description
POST	/v2.1/{tenant_id}/os-certificates	Creates a certificate.

Normal response codes: 200

4.28.1.1. Request

This table shows the URI parameters for the create certificate request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.28.1.2. Response

Example 4.160. Create certificate: JSON response

```
{
  "certificate": {
    "data": "Certificate:\n      Data:\n      Version: 1 (0x0)\n
Serial Number: 1018 (0x3fa)\n      Signature Algorithm: md5WithRSAEncryption
\n      Issuer: O=NOVA ROOT, L=Mountain View, ST=California, C=US\n
Validity\n      Not Before: Aug 12 07:20:30 2013 GMT\n
Not After : Aug 12 07:20:30 2014 GMT\n      Subject: C=US, ST=
California, O=OpenStack, OU=NovaDev, CN=openstack-fake-2013-08-12T07:20:30Z
\n      Subject Public Key Info:\n      Public Key Algorithm:
rsaEncryption\n      Public-Key: (1024 bit)\n
Modulus:\n      00:ac:ff:b1:d1:ed:54:4e:35:6c:34:b4:8f:0b:04:
\n      50:25:a3:e2:4f:02:4c:4f:26:59:bd:f3:fd:eb:da:
\n      18:c2:36:aa:63:42:72:1f:88:4f:3a:ec:e7:9f:8e:
\n      44:2a:d3:b8:94:7b:20:41:f8:48:02:57:91:4c:16:
\n      62:f1:21:d4:f2:40:b5:86:50:d9:61:f0:be:ff:d8:
\n      8d:9f:4b:aa:6a:07:38:a2:7f:87:21:fc:e6:6e:1d:
\n      0a:95:1a:90:0e:60:c2:24:e9:8e:e8:68:1b:e9:f3:
\n      c6:b0:7c:da:c5:20:66:9b:85:ea:f5:c9:a7:de:ee:
\n      16:b1:51:a0:4d:e3:95:98:df\n
Exponent: 65537 (0x10001)\n      Signature Algorithm: md5WithRSAEncryption
\n      15:42:ca:71:cc:32:af:dc:cf:45:91:df:8a:b8:30:c4:7f:78:
\n      80:a7:25:c2:d9:81:3e:b3:dd:22:cc:3b:f8:94:e7:8f:04:f6:
\n      93:04:9e:85:d4:10:40:ff:5a:07:47:24:b5:ae:93:ad:8d:e1:
\n      e6:54:4a:8d:4a:29:53:c4:8d:04:6b:0b:f6:af:38:78:02:c5:
\n      05:19:89:82:2d:ba:fd:11:3c:1e:18:c9:0c:3d:03:93:6e:bc:
\n      66:70:34:ee:03:78:8a:1d:3d:64:e8:20:2f:90:81:8e:49:1d:
\n      07:37:15:66:42:cb:58:39:ad:56:ce:ed:47:c6:78:0b:0e:75:
\n      29:ca\n-----BEGIN CERTIFICATE-----\n
nMIICNDCCAZ0CAgP6MA0GCSqGSIb3DQEBBAUAME4xEjAQBgNVBAoTCU5PVkEgUk9P\
nVDEWMBQGA1UEBxMNTW91bnRhaw4gVmllldzETMBEGA1UECBMKQ2FsaWZvcn5pYTEL\
nMAKGA1UEBhMCVVmwHhcNMTMwODEyMDcyMDMwHhcNMTQwODEyMDcyMDMwWjB2MQsw\
nCQYDVQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcn5pYTESMBAGA1UECgwJT3Blb1N0\
nYWNRMRADgYDVQQQLDAdOb3ZzRGV2MSwwKgYDVQQDDCNvcGVuc3RhY2stZmFrZS0y\
nMDEzLTA4LTEyVDA3OjIwOjMwWjCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA\
nrP+x0e1UTjVsNLSPCwRQJaPiTwJMTyZZvfp969oYwjaqY0JyH4hPOuznn45EkT04\
```



```

nLHsgQfhIAleRTBZi8SHU8kC1hLDZYfC+/9iNn0uqagc4on+HIfzmbh0KlRqQDmDC\
nJ0m06Ggb6fPGsHzaxSBmm4Xq9cmn3u4WsVGgTe0VmN8CAwEAATANBgkqhkiG9w0B\
nAQQFAA0BgQAVQspxzDKv2.1M9Fkd+KuDDef3iApyXC2YE+s90izDv4l0ePBPATBJ6F
\n1BBA/1oHRYs1rp0tjeHmVEqNSilTxI0Eawv2rzh4AsUFGYmCLbr9ETweGMkMPQOT\
nbrxmcDTuA3iKHT1k6CAvkIG0SR0HNxVmQstY0a1Wzu1HxngLDnUpyg==\n-----END
CERTIFICATE-----\n",
    "private_key": "-----BEGIN RSA PRIVATE KEY-----\
nMIICXgIBAAKBgQCs/7HR7VR0Nww0tI8LBFAlo+JPAkxPJlm98/3r2hjCNqpjQnIf\
niE8670efjkQq07iUeyBB+EgCV5FMFmLxIdTyQLWGUNlh8L7/2I2fS6pqBzii4ch\
n/OZuHQqVGpAOYMIk6Y7oaBvp88awfNrFIGabher1yafe7haxUaBN45WY3wIDAQAB\
nAoGBAircr2I/KyWf0hw4Nn10V9TuyE/9Gz2JHg3QFKjFJox2DqygADT5WAeHc6Bq\
nNKNf0NA2SL1LSpm+ql01tv0w4Vje5TF60HiIzHuTTnXggG6vuA8rxp6L24HtkAcC\
n0CBno9ggSX6jVornJPBfxpkwITYSvH57BUFVD7ovbPyWGzS5AkEA1JeUtL6zxwps\
nWRr1aJ8Ill2uQk/RUIvSZOU61s+B190zvHikFy8LD8CI6vvBmjC/IZuZVedufjqs\
n4vX82uD03QJBANBSH2b2dyB4AGVFY9vXMrtALAspJHbLHy+zTKxlGPFiuz7Se3ps\
n8Kehz4C/CBXgQkk194dwFSGE19/PQfyJR0sCQCFFDJZhrBtUMwMZ2zSRiN5BUGt\
nbwuncS+0S1Su3Yz5VRYq2BZYEPHKtYrAFkLWQ8eRwTaWaN5pFE/fb380gQXdAkA4\
nDm0W/K0zlhbuyUxEpNQ28/6mBi0ktiWvLT0tioq6sYmXLwZA/D2JrhXrG/xt/ol3\
nr8jqrfrNRsLByLhAgh0N/AkEAl2eR0097lTEgFNqzIQwVmIAN9mB03cnf3tycvlDU\
nm6eb2CS242y4QalfCCAEjxoJURdfsm3/D1iFo00X+IwF+A==\n-----END RSA PRIVATE
KEY-----\n"
    }
}

```

4.28.2. Show certificate details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-certificates/{certificate_id}	Shows details for a certificate.

Normal response codes: 200

4.28.2.1. Request

This table shows the URI parameters for the show certificate details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{certificate_id}	String	The certificate ID, as a string. The only valid value is root.

This operation does not accept a request body.

4.28.2.2. Response

Example 4.161. Show certificate details: JSON response

```
{
  "certificate": {
    "data": "-----BEGIN CERTIFICATE-----\n
nMIICyzCCAjSgAwIBAgIJAj8zSIxUp/m4MA0GCSqGSIb3DQEBAUAME4xEjAQBgNV\
nBAoTCU5PVkEgUk9PVDEWMBQGA1UEBxMNTW91bnRhaw4gVmllldzETMBEGA1UECBMK\
nQ2FsaWZvcn5pYTELMAkGA1UEBhMCVVMwHhcNMTIxMDEzMDEzMzUwHcNMTMxMDE3\
nMDEzMzUwZjBOMRIwEAYDVQQKEwlOT1ZBIFJPT1QxYjAUBgNVBACjTDU1vdW50YWlu\
nIFZpZXcxZzARBgNVBAGTCkNhbgGmb3JuawExCzAJBgNVBAYTAlVTMIGfMA0GCSqG\
nSIb3DQEBAQUAA4GNADCBiQKBgQDXW4QfQQxJG4MqurqK8nU/Lge0mfNKxXj/Gwvg\
n2sQVwxzmKfoxih8Nn6yt0yHMNjhoji1UowI03TXUnPZRAZmsypGKZeBd7Y1Z0CPB\
nXGZVGrQm+PB2kZU+3cd8fVKcueMLLeZ+LRt5d0njnoKhc5xjqMlFFPimHMba40L6\
nTnYzPQIDAQABo4GwMIGtMAwGA1UdEwQFMAMBAf8wHQYDVR00BBYEFKyoKu4SM0FM\
ngx5Ec7p0nrCkabvxxMH4GA1UdIwR3MHWAFKyoKu4SM0FMgx5Ec7p0nrCkabvxoVKK\
nUDBOMRIwEAYDVQQKEwlOT1ZBIFJPT1QxYjAUBgNVBACjTDU1vdW50YWluIFZpZXcx\
nEzARBgNVBAGTCkNhbgGmb3JuawExCzAJBgNVBAYTAlVTggkAnzNIjFSn+bgwDQYJ\
nKoZiHvcNAQEEBQADgYEAXuvXlu1o/SVvykSLhHW8QiAY00yzN/eDzYmZGomgiuo0\n/
x+ayVzbrz1UWZnBD+lc4hll2iELSmf22LjLoF+s/9NyPqHxGL3FrfatBkndaiF8\nAx/\
TMEyCPL7IQWi+3zzatqOKHSHiG7a9SGn/7o2aNTIWKVulfy5GvmbBjBM/0UE=\n
-----END
CERTIFICATE-----\n",
    "private_key": null
  }
}
```

4.29. Cloudpipe (os-cloudpipe)

Manages virtual VPNs for projects.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-cloudpipe	Lists cloudpipes.
POST	/v2.1/{tenant_id}/os-cloudpipe	Creates a cloudpipe.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-cloud-pipe/configure-project	Updates the virtual private network (VPN) IP address and port for a cloudpipe instance.

4.29.1. List cloudpipes

Method	URI	Description
GET	/v2.1/{tenant_id}/os-cloudpipe	Lists cloudpipes.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.29.1.1. Request

This table shows the URI parameters for the list cloudpipes request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.29.1.2. Response

Example 4.162. List cloudpipes: JSON response

```
{
  "cloudpipes": [
    {
      "created_at": "2012-11-27T17:18:01Z",
      "instance_id": "27deecdb-baa3-4a26-9c82-32994b815b01",
      "internal_ip": "192.168.0.3",
      "project_id": "fa1765bd-a352-49c7-a6b7-8ee108a3cb0c",
      "public_ip": "127.0.0.1",
      "public_port": 22,
      "state": "down"
    }
  ]
}
```

4.29.2. Create cloudpipe

Method	URI	Description
POST	/v2.1/{tenant_id}/os-cloudpipe	Creates a cloudpipe.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.29.2.1. Request

This table shows the URI parameters for the create cloudpipe request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.163. Create cloudpipe: JSON request

```
{
  "cloudpipe": {
    "project_id": "059f21e3-c20e-4efc-9e7a-eba2ab3c6f9a"
  }
}
```

This operation does not accept a request body.

4.29.2.2. Response

Example 4.164. Create cloudpipe: JSON response

```
{
  "instance_id": "1e9b8425-34af-488e-b969-4d46f4a6382e"
}
```

4.29.3. Update cloudpipe

Method	URI	Description
POST	/v2.1/{tenant_id}/os-cloud-pipe/configure-project	Updates the virtual private network (VPN) IP address and port for a cloudpipe instance.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), badMediaType (415), NetworkNotFound (400)

4.29.3.1. Request

This table shows the URI parameters for the update cloudpipe request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.165. Update cloudpipe: JSON request

```
{
  "configure_project": {
    "vpn_ip": "192.168.1.1",
    "vpn_port": "2000"
  }
}
```

This operation does not accept a request body.

4.30. Server consoles (servers, os-consoles, os-console-auth-token)

Manages server consoles.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/consoles	Creates a console for a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/consoles	Lists all consoles for a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/consoles/{console_id}	Shows details for a console for a server instance.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/consoles/{console_id}	Deletes a console for a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-console-auth-token	Shows the authentication token for a console for a server instance.

4.30.1. Create console

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/consoles	Creates a console for a server instance.

Normal response codes: 200

4.30.1.1. Request

This table shows the URI parameters for the create console request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.30.2. Lists consoles

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/consoles	Lists all consoles for a server instance.

Normal response codes: 200

4.30.2.1. Request

This table shows the URI parameters for the lists consoles request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.30.2.2. Response

Example 4.166. List consoles: JSON response

```
{
  "consoles": [
    {
      "console": {
        "console_type": "fake",
        "id": 1
      }
    }
  ]
}
```


4.30.3. Show console details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/consoles/{console_id}	Shows details for a console for a server instance.

Normal response codes: 200

4.30.3.1. Request

This table shows the URI parameters for the show console details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{console_id}	UUID	The UUID of the console.

This operation does not accept a request body.

4.30.3.2. Response

Example 4.167. List consoles: JSON response

```
{
  "consoles": [
    {
      "console": {
        "console_type": "fake",
        "id": 1
      }
    }
  ]
}
```

4.30.4. Delete console

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}/consoles/{console_id}	Deletes a console for a server instance.

Normal response codes: 202

4.30.4.1. Request

This table shows the URI parameters for the delete console request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.
{console_id}	UUID	The UUID of the console.

This operation does not accept a request body.

4.30.5. Show console authentication token

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-console-auth-token	Shows the authentication token for a console for a server instance.

This feature is available for rdp-html5 console type only.

Normal response codes: 200

4.30.5.1. Request

This table shows the URI parameters for the show console authentication token request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.30.5.2. Response

Example 4.168. List consoles: JSON response

```
{
  "console": {
    "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
    "host": "localhost",
    "port": 5900,
    "internal_access_path": "51af38c3-555e-4884-a314-6c8cdde37444"
  }
}
```

4.31. Fixed IPs (os-fixed-ips)

Shows data for a fixed IP, such as host name, CIDR, and address. Also, reserves and frees a fixed IP address.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}	Shows details for a fixed IP address.
POST	/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}/action	Reserves or releases a fixed IP.

4.31.1. Show fixed IP details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}	Shows details for a fixed IP address.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.31.1.1. Request

This table shows the URI parameters for the show fixed ip details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{fixed_ip}	String	The fixed IP of interest to you.

This operation does not accept a request body.

4.31.1.2. Response

Example 4.169. Show fixed IP details: JSON response

```
{
  "fixed_ip": {
    "address": "192.168.1.1",
    "cidr": "192.168.1.0/24",
    "host": "host",
    "hostname": "openstack"
  }
}
```

4.31.2. Reserve or release a fixed IP

Method	URI	Description
POST	/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}/action	Reserves or releases a fixed IP.

To reserve a fixed IP address, specify `reserve` in the request body. To release a fixed IP address, specify `unreserve` in the request body.

Normal response codes: 202

Error response codes: `computeFault` (400, 500, ...), `serviceUnavailable` (503), `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `badMediaType` (415), `NetworkNotFound` (400)

4.31.2.1. Request

This table shows the URI parameters for the reserve or release a fixed ip request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{fixed_ip}	String	The fixed IP of interest to you.

Example 4.170. Reserve or release a fixed IP: JSON request

```
{
  "reserve": null
}
```

4.32. Floating IP DNS records (os-floating-ip-dns)

Manages DNS records associated with floating IP addresses. The API dispatches requests to a DNS driver that is selected at startup.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ip-dns	Lists registered DNS domains published by the DNS drivers.
PUT	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}	Creates or updates a DNS domain.
DELETE	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}	Deletes a DNS domain and all associated host entries.
PUT	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Creates or updates a DNS entry.
GET	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Finds a unique DNS entry for a domain and name.
DELETE	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Deletes a DNS entry.
GET	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}	Lists DNS entries for a domain and IP.

4.32.1. List DNS domains

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ip-dns	Lists registered DNS domains published by the DNS drivers.

Normal response codes: 200

4.32.1.1. Request

This table shows the URI parameters for the list dns domains request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.32.1.2. Response

Example 4.171. List DNS domains: JSON response

```
{
  "domain_entries": [
    {
      "availability_zone": null,
      "domain": "domain1.example.org",
      "project": "project1",
      "scope": "public"
    }
  ]
}
```

4.32.2. Create or update DNS domain

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}	Creates or updates a DNS domain.

Normal response codes: 200

4.32.2.1. Request

This table shows the URI parameters for the create or update dns domain request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{domain}	String	The registered DNS domain that the DNS drivers publish.

Example 4.172. Create or update DNS domain: JSON request

```
{
  "domain_entry": {
    "scope": "public",
    "project": "project1"
  }
}
```

4.32.2.2. Response

Example 4.173. Create or update DNS domain: JSON response

```
{
  "domain_entry": {
    "availability_zone": null,
    "domain": "domain1.example.org",
    "project": "project1",
    "scope": "public"
  }
}
```

4.32.3. Delete DNS domain

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}	Deletes a DNS domain and all associated host entries.

Normal response codes: 200

4.32.3.1. Request

This table shows the URI parameters for the delete dns domain request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{domain}	String	The registered DNS domain that the DNS drivers publish.

This operation does not accept a request body.

4.32.4. Create or update DNS entry

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Creates or updates a DNS entry.

Normal response codes: 200

4.32.4.1. Request

This table shows the URI parameters for the create or update dns entry request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{domain}	String	The registered DNS domain that the DNS drivers publish.
{name}	String	The name of the DNS entry.

Example 4.174. Create or update DNS entry: JSON request

```
{
  "dns_entry": {
    "ip": "192.168.53.11",
    "dns_type": "A"
  }
}
```

4.32.4.2. Response

Example 4.175. Create or update DNS entry: JSON response

```
{
  "dns_entry": {
    "domain": "domain1.example.org",
    "id": null,
    "ip": "192.168.1.1",
    "name": "instance1",
    "type": "A"
  }
}
```

4.32.5. Find unique DNS entry

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Finds a unique DNS entry for a domain and name.

Normal response codes: 200

4.32.5.1. Request

This table shows the URI parameters for the find unique dns entry request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{domain}	String	The registered DNS domain that the DNS drivers publish.
{name}	String	The name of the DNS entry.

This operation does not accept a request body.

4.32.5.2. Response

Example 4.176. Find unique DNS entry: JSON response

```
{
  "dns_entry": {
    "domain": "domain1.example.org",
    "id": null,
    "ip": "192.168.1.1",
    "name": "instance1",
    "type": null
  }
}
```

4.32.6. Delete DNS entry

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Deletes a DNS entry.

Normal response codes: 200

4.32.6.1. Request

This table shows the URI parameters for the delete dns entry request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{domain}	String	The registered DNS domain that the DNS drivers publish.
{name}	String	The name of the DNS entry.

This operation does not accept a request body.

4.32.7. List DNS entries

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}	Lists DNS entries for a domain and IP.

Normal response codes: 200

4.32.7.1. Request

This table shows the URI parameters for the list dns entries request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{domain}	String	The registered DNS domain that the DNS drivers publish.
{ip}	String	The IP address.

This operation does not accept a request body.

4.32.7.2. Response

Example 4.177. List DNS entries: JSON response

```
{
  "dns_entries": [
    {
      "domain": "domain1.example.org",
      "id": null,
      "ip": "192.168.1.1",
      "name": "instance1",
      "type": null
    }
  ]
}
```

4.33. Floating IP pools (os-floating-ip-pools)

Manages groups of floating IPs.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ip-pools	Lists floating IP pools.

4.33.1. List floating IP pools

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ip-pools	Lists floating IP pools.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.33.1.1. Request

This table shows the URI parameters for the list floating ip pools request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.33.1.2. Response

Example 4.178. List floating IP pools: JSON response

```
{
  "floating_ip_pools": [
    {
      "name": "pool1"
    },
    {
      "name": "pool2"
    }
  ]
}
```

4.34. Floating IPs (os-floating-ips)

Lists floating IP addresses for a project. Also, creates (allocates) a floating IP address for a project, shows floating IP address details, and deletes (deallocates) a floating IP address from a project.

The cloud administrator configures a pool of floating IP addresses in OpenStack Compute. The project quota defines the maximum number of floating IP addresses that you can allocate to the project. After you [allocate a floating IP address](#) for a project, you can:

- [Add \(associate\) the floating IP address](#) with an instance in the project. You can associate only one floating IP address with an instance at a time.
- [Remove \(disassociate\) the floating IP address](#) from an instance in the project.
- Delete, or deallocate, a floating IP from the project, which automatically deletes any associations for that IP address.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ips	Lists floating IP addresses associated with the tenant or account.
POST	/v2.1/{tenant_id}/os-floating-ips	Creates, or allocates, a floating IP address for the current project. By default, the floating IP address is allocated from the public pool.
GET	/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}	Shows details for a floating IP address, by ID, that is associated with the tenant or account.
DELETE	/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}	Deletes, or deallocates, a floating IP address from the current project and returns it to the pool from which it was allocated.

4.34.1. List floating IP addresses

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ips	Lists floating IP addresses associated with the tenant or account.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.34.1.1. Request

This table shows the URI parameters for the list floating ip addresses request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.34.1.2. Response

Example 4.179. List floating IP addresses: JSON response

```
{
  "floatingips": [
    {
      "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
      "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
      "floating_network_id": "376da547-b977-4cfe-9cba-275c80deb57",
      "fixed_ip_address": "10.0.0.3",
      "floating_ip_address": "172.24.4.228",
      "port_id": "ce705c24-c1ef-408a-bda3-7bbd946164ab",
      "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7",
      "status": "ACTIVE"
    },
    {
      "router_id": null,
      "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
      "floating_network_id": "376da547-b977-4cfe-9cba-275c80deb57",
      "fixed_ip_address": null,
      "floating_ip_address": "172.24.4.227",
      "port_id": null,
      "id": "61cea855-49cb-4846-997d-801b70c71bdd",
      "status": "DOWN"
    }
  ]
}
```

4.34.2. Create (allocate) floating IP address

Method	URI	Description
POST	/v2.1/{tenant_id}/os-floating-ips	Creates, or allocates, a floating IP address for the current project. By default, the floating IP address is allocated from the public pool.

If more than one floating IP address pool is available, use the `pool` parameter to specify from which pool to allocate the IP address.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

Error response codes: 400

4.34.2.1. Request

This table shows the URI parameters for the create (allocate) floating ip address request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.180. Create (allocate) floating IP address: JSON request

```
{
  "pool": "public"
}
```

4.34.2.2. Response

Example 4.181. Create (allocate) floating IP address: JSON response

```
{
  "floating_ip": {
    "instance_id": null,
    "ip": "172.24.4.4",
    "fixed_ip": null,
    "id": "c9c04158-3ed4-449c-953a-aa21fb47cde7",
    "pool": "public"
  }
}
```


4.34.3. Show floating IP address details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}	Shows details for a floating IP address, by ID, that is associated with the tenant or account.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.34.3.1. Request

This table shows the URI parameters for the show floating ip address details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{floating_ip_id}	Int	The ID of the floating IP address.

This operation does not accept a request body.

4.34.3.2. Response

Example 4.182. Show floating IP address details: JSON response

```
{
  "floating_ip": {
    "instance_id": null,
    "ip": "172.24.4.3",
    "fixed_ip": null,
    "id": "b310fff3-c467-4950-9b00-038afebd151c",
    "pool": "public"
  }
}
```

4.34.4. Delete (deallocate) floating IP address

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}	Deletes, or deallocates, a floating IP address from the current project and returns it to the pool from which it was allocated.

If the IP address is still associated with a running instance, it is automatically disassociated from that instance.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.34.4.1. Request

This table shows the URI parameters for the delete (deallocate) floating ip address request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{floating_ip_id}	Int	The ID of the floating IP address.

This operation does not accept a request body.

4.35. Floating IPs bulk (os-floating-ips-bulk)

(nova-network only) Bulk-creates, deletes, and lists floating IPs. Default pool name is nova.

To view available pools, use the `os-floating-ip-pools` extension.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ips-bulk	Lists all floating IPs.
POST	/v2.1/{tenant_id}/os-floating-ips-bulk	Bulk-creates floating IPs.
POST	/v2.1/{tenant_id}/os-floating-ips-bulk/delete	Bulk-deletes floating IPs.
GET	/v2.1/{tenant_id}/os-floating-ips-bulk/{host_name}	Lists all floating IPs for a host.

4.35.1. List floating IPs

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ips-bulk	Lists all floating IPs.

Normal response codes: 200

4.35.1.1. Request

This table shows the URI parameters for the list floating ips request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.35.1.2. Response

Example 4.183. List floating IPs: JSON response

```
{
  "floating_ip_info": [
    {
      "address": "10.10.10.1",
      "instance_uuid": null,
      "fixed_ip": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    },
    {
      "address": "10.10.10.2",
      "instance_uuid": null,
      "fixed_ip": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    },
    {
      "address": "10.10.10.3",
      "instance_uuid": null,
      "fixed_ip": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    }
  ]
}
```

4.35.2. Create floating IPs

Method	URI	Description
POST	/v2.1/{tenant_id}/os-floating-ips-bulk	Bulk-creates floating IPs.

Normal response codes: 200

4.35.2.1. Request

This table shows the URI parameters for the create floating ips request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.184. Create floating IPs: JSON request

```
{
  "floating_ips_bulk_create": {
    "ip_range": "192.168.1.0/24",
    "pool": "nova",
    "interface": "eth0"
  }
}
```

4.35.2.2. Response

Example 4.185. Create floating IPs: JSON response

```
{
  "floating_ips_bulk_create": {
    "interface": "eth0",
    "ip_range": "192.168.1.0/24",
    "pool": "nova"
  }
}
```

4.35.3. Bulk-delete floating IPs

Method	URI	Description
POST	/v2.1/{tenant_id}/os-floating-ips-bulk/delete	Bulk-deletes floating IPs.

Normal response codes: 200

4.35.3.1. Request

This table shows the URI parameters for the bulk-delete floating ips request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.186. Bulk-delete floating IPs: JSON request

```
{
  "ip_range": "192.168.1.0/24"
}
```

4.35.3.2. Response

Example 4.187. Bulk-delete floating IPs: JSON response

```
{
  "floating_ips_bulk_delete": "192.168.1.0/24"
}
```

4.35.4. List floating IPs by host

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ips-bulk/{host_name}	Lists all floating IPs for a host.

Normal response codes: 200

4.35.4.1. Request

This table shows the URI parameters for the list floating ips by host request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{host_name}	String	The name of the host.

This operation does not accept a request body.

4.35.4.2. Response

Example 4.188. List floating IPs by host: JSON response

```
{
  "floating_ip_info": [
    {
      "address": "10.10.10.3",
      "instance_uuid": null,
      "fixed_ip": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    }
  ]
}
```

4.36. Ping instances (os-fping)

Pings instances and reports which ones are alive.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-fping{?all_tenants,include,exclude}	Run the fping utility to ping instances and report which ones are alive.
GET	/v2.1/{tenant_id}/os-fping/{instance_id}	Run the fping utility to ping an instance and report whether it is alive.

4.36.1. Ping instances

Method	URI	Description
GET	/v2.1/{tenant_id}/os-fping{?all_tenants,include,exclude}	Run the fping utility to ping instances and report which ones are alive.

Specify the `all_tenants=1` query parameter to ping instances for all tenants. For example:

```
GET /os-fping?all_tenants=1
```

Specify the `include` and `exclude` query parameters to filter the results. For example:

```
GET /os-fping?all_tenants=1&include=uuid1,uuid2&exclude=uuid3,uuid4
```

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.36.1.1. Request

This table shows the URI parameters for the ping instances request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the ping instances request:

Name	Type	Description
all_tenants	Int (Optional)	Specify the <code>all_tenants=1</code> query parameter to ping instances for all tenants.
include	String (Optional)	Specify <code>include=uuid[, uuid...]</code> to include the instances in the results.
exclude	String (Optional)	Specify <code>exclude=uuid[, uuid...]</code> to exclude the instances from the results.

This operation does not accept a request body.

4.36.1.2. Response

Example 4.189. Ping instances: JSON response

```
{
  "servers": [
    {
      "alive": false,
      "id": "1d1aea35-472b-40cf-9337-8eb68480aaa1",
      "project_id": "openstack"
    }
  ]
}
```

```
    ]  
}
```


4.36.2. Ping an instance

Method	URI	Description
GET	/v2.1/{tenant_id}/os-fping/{instance_id}	Run the fping utility to ping an instance and report whether it is alive.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.36.2.1. Request

This table shows the URI parameters for the ping an instance request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{instance_id}	UUID	The UUID of the instance.

This operation does not accept a request body.

4.36.2.2. Response

Example 4.190. Ping an instance: JSON response

```
{
  "server": {
    "alive": false,
    "id": "f5e6fd6d-c0a3-4f9e-aabf-d69196b6d11a",
    "project_id": "openstack"
  }
}
```

4.37. Hosts (os-hosts)

Manages physical hosts. Some virt drivers don not support all host functions. For more information, see [nova virt support matrix](#)

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts	Lists hosts.
PUT	/v2.1/{tenant_id}/os-hosts/{host_name}	Enables or puts a host in maintenance mode.
GET	/v2.1/{tenant_id}/os-hosts/{host_name}	Shows details for a host.
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/reboot	Reboots a host.
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/shutdown	Shuts down a host.
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/startup	Starts a host.

4.37.1. List hosts

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts	Lists hosts.

Normal response codes: 200

4.37.1.1. Request

This table shows the URI parameters for the list hosts request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.37.1.2. Response

Example 4.191. List hosts: JSON response

```
{
  "hosts": [
    {
      "host_name": "b6e4adbc193d428ea923899d07fb001e",
      "service": "conductor",
      "zone": "internal"
    },
    {
      "host_name": "09c025b0efc64211bd23fc50fa974cdf",
      "service": "compute",
      "zone": "nova"
    },
    {
      "host_name": "a942ebfa00064d9d89a9e5a175cb9ba8",
      "service": "cert",
      "zone": "internal"
    },
    {
      "host_name": "e73ec0bd35c64de4a1adfa8b8969a1f6",
      "service": "consoleauth",
      "zone": "internal"
    },
    {
      "host_name": "396a8a0a234f476eb05fb9fbc5802ba7",
      "service": "network",
      "zone": "internal"
    },
    {
      "host_name": "abffda96592c4eacaf4111c28fddee17",
      "service": "scheduler",
      "zone": "internal"
    },
    {
      "host_name": "a8820f04962a4b4ba9fe2e9540c24094",
      "service": "cells",

```

```
        "zone": "internal"  
      }  
    ]  
  }
```

4.37.2. Enable host

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-hosts/{host_name}	Enables or puts a host in maintenance mode.

Normal response codes: 200

4.37.2.1. Request

This table shows the URI parameters for the enable host request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{host_name}	String	The name of the host.

Example 4.192. Enable host: JSON request

```
{
  "status": "enable",
  "maintenance_mode": "disable"
}
```

4.37.2.2. Response

Example 4.193. Enable host: JSON response

```
{
  "host": "65c5d5b7e3bd44308e67fc50f362aee6",
  "maintenance_mode": "off_maintenance",
  "status": "enabled"
}
```

4.37.3. Show host details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts/{host_name}	Shows details for a host.

Normal response codes: 200

4.37.3.1. Request

This table shows the URI parameters for the show host details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{host_name}	String	The name of the host.

This operation does not accept a request body.

4.37.3.2. Response

Example 4.194. Show host details: JSON response

```
{
  "host": [
    {
      "resource": {
        "cpu": 1,
        "disk_gb": 1028,
        "host": "c1a7de0ac9d94e4baceae031d05caae3",
        "memory_mb": 8192,
        "project": "(total)"
      }
    },
    {
      "resource": {
        "cpu": 0,
        "disk_gb": 0,
        "host": "c1a7de0ac9d94e4baceae031d05caae3",
        "memory_mb": 512,
        "project": "(used_now)"
      }
    },
    {
      "resource": {
        "cpu": 0,
        "disk_gb": 0,
        "host": "c1a7de0ac9d94e4baceae031d05caae3",
        "memory_mb": 0,
        "project": "(used_max)"
      }
    }
  ]
}
```

4.37.4. Reboot host

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/reboot	Reboots a host.

Normal response codes: 200

4.37.4.1. Request

This table shows the URI parameters for the reboot host request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{host_name}	String	The name of the host.

This operation does not accept a request body.

4.37.4.2. Response

Example 4.195. Reboot host: JSON response

```
{
  "host": "9557750dbc464741a89c907921c1cb31",
  "power_action": "reboot"
}
```

4.37.5. Shut down host

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/shutdown	Shuts down a host.

Normal response codes: 200

4.37.5.1. Request

This table shows the URI parameters for the shut down host request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{host_name}	String	The name of the host.

This operation does not accept a request body.

4.37.5.2. Response

Example 4.196. Shut down host: JSON response

```
{
  "host": "77cfa0002e4d45fe97f185968111b27b",
  "power_action": "shutdown"
}
```


4.37.6. Start host

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/startup	Starts a host.

Normal response codes: 200

4.37.6.1. Request

This table shows the URI parameters for the start host request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{host_name}	String	The name of the host.

This operation does not accept a request body.

4.37.6.2. Response

Example 4.197. Start host: JSON response

```
{
  "host": "4b392b27930343bbaa27fd5d8328a564",
  "power_action": "startup"
}
```

4.38. Hypervisors (os-hypervisors)

Lists all hypervisors, shows summary statistics for all hypervisors over all compute nodes, shows details for a hypervisor, and shows the uptime for a hypervisor.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hypervisors	Lists hypervisors.
GET	/v2.1/{tenant_id}/os-hypervisors/statistics	Shows summary statistics for all hypervisors over all compute nodes.
GET	/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}	Shows details for a hypervisor.
GET	/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}/uptime	Shows the uptime for a hypervisor.

4.38.1. List hypervisors

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hypervisors	Lists hypervisors.

Normal response codes: 200

4.38.1.1. Request

This table shows the URI parameters for the list hypervisors request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.38.1.2. Response

Example 4.198. List hypervisors: JSON response

```
{
  "hypervisors": [
    {
      "status": "enabled",
      "state": "up",
      "id": 1,
      "hypervisor_hostname": "fake-mini"
    }
  ]
}
```

4.38.2. Show hypervisor statistics

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hypervisors/statistics	Shows summary statistics for all hypervisors over all compute nodes.

Normal response codes: 200

4.38.2.1. Request

This table shows the URI parameters for the show hypervisor statistics request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.38.2.2. Response

Example 4.199. Show hypervisor statistics: JSON response

```
{
  "hypervisor_statistics": {
    "count": 1,
    "vcpus_used": 0,
    "local_gb_used": 0,
    "memory_mb": 7980,
    "current_workload": 0,
    "vcpus": 8,
    "running_vms": 0,
    "free_disk_gb": 157,
    "disk_available_least": 140,
    "local_gb": 157,
    "free_ram_mb": 7468,
    "memory_mb_used": 512
  }
}
```

4.38.3. Show hypervisor details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}	Shows details for a hypervisor.

Normal response codes: 200

4.38.3.1. Request

This table shows the URI parameters for the show hypervisor details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{hypervisor_id}	UUID	The UUID of the hypervisor.

This operation does not accept a request body.

4.38.3.2. Response

Example 4.200. Show hypervisor details: JSON response

```
{
  "hypervisor": {
    "status": "enabled",
    "service": {
      "host": "fake-mini",
      "disabled_reason": null,
      "id": 6
    },
    "vcpus_used": 0,
    "hypervisor_type": "QEMU",
    "local_gb_used": 0,
    "vcpus": 8,
    "hypervisor_hostname": "fake-mini",
    "memory_mb_used": 512,
    "memory_mb": 7980,
    "current_workload": 0,
    "state": "up",
    "host_ip": "23.253.248.171",
    "cpu_info": "{\n\"vendor\": \"Intel\", \"model\": \"gate64\", \"arch\": \"x86_64\", \"features\": [\"pge\", \"clflush\", \"sep\", \"syscall\", \"vme\", \"msr\", \"cmov\", \"fpu\", \"pat\", \"lm\", \"tsc\", \"nx\", \"fxsr\", \"sse4.1\", \"pae\", \"sse4.2\", \"pclmuldq\", \"tsc-deadline\", \"mmx\", \"cx8\", \"mce\", \"de\", \"rdtscp\", \"mca\", \"pse\", \"pni\", \"popcnt\", \"apic\", \"sse\", \"lahf_lm\", \"aes\", \"sse2\", \"hypervisor\", \"ssse3\", \"cx16\", \"mtrr\", \"x2apic\"], \"topology\": {\"cores\": 1, \"cells\": 1, \"threads\": 1, \"sockets\": 8}}",
    "running_vms": 0,
    "free_disk_gb": 157,
    "hypervisor_version": 2000000,
    "disk_available_least": 140,
    "local_gb": 157,
    "free_ram_mb": 7468,
    "id": 1
  }
}
```

```
}  
}
```

4.38.4. Show hypervisor uptime

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}/uptime	Shows the uptime for a hypervisor.

Normal response codes: 200

4.38.4.1. Request

This table shows the URI parameters for the show hypervisor uptime request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{hypervisor_id}	UUID	The UUID of the hypervisor.

This operation does not accept a request body.

4.38.4.2. Response

Example 4.201. Show hypervisor uptime: JSON response

```
{
  "hypervisor": {
    "status": "enabled",
    "state": "up",
    "id": 1,
    "hypervisor_hostname": "fake-mini",
    "uptime": " 16:09:43 up 8 days, 19:58,  1 user,  load average: 0.86,
0.63, 0.55\n"
  }
}
```

4.39. Instance usage audit log (os-instance-usage-audit-log)

Administrator only. Monitors task logs.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-instance_usage_audit_log	Lists usage audits for an instance.
GET	/v2.1/{tenant_id}/os-instance_usage_audit_log/{before_timestamp}{?before_timestamp}	Lists usage audits that occurred before a specified time.

4.39.1. List usage audits for an instance

Method	URI	Description
GET	/v2.1/{tenant_id}/os-instance_usage_audit_log	Lists usage audits for an instance.

Normal response codes: 200

4.39.1.1. Request

This table shows the URI parameters for the list usage audits for an instance request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.39.1.2. Response

Example 4.202. List usage audits for an instance: JSON response

```
{
  "instance_usage_audit_logs": {
    "hosts_not_run": [
      "f4eb7cfd155f4574967f8b55a7faed75"
    ],
    "log": {},
    "num_hosts": 1,
    "num_hosts_done": 0,
    "num_hosts_not_run": 1,
    "num_hosts_running": 0,
    "overall_status": "0 of 1 hosts done. 0 errors.",
    "period_beginning": "2012-12-01 00:00:00",
    "period_ending": "2013-01-01 00:00:00",
    "total_errors": 0,
    "total_instances": 0
  }
}
```

4.39.2. List usage audits before specified time

Method	URI	Description
GET	/v2.1/{tenant_id}/os-instance_usage_audit_log/{before_timestamp}{?before_timestamp}	Lists usage audits that occurred before a specified time.

Normal response codes: 200

4.39.2.1. Request

This table shows the URI parameters for the list usage audits before specified time request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list usage audits before specified time request:

Name	Type	Description
before_timestamp	DateTime (Required)	Filters the response by the date and time before which to list usage audits. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, returns the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. If you omit the time zone, the UTC time zone is assumed.

This operation does not accept a request body.

4.39.2.2. Response

Example 4.203. List usage audits before specified time: JSON response

```
{
  "instance_usage_audit_log": {
    "hosts_not_run": [
      "8e33da2b48684ef3ab165444d6a7384c"
    ],
    "log": {},
    "num_hosts": 1,
    "num_hosts_done": 0,
    "num_hosts_not_run": 1,
    "num_hosts_running": 0,
    "overall_status": "0 of 1 hosts done. 0 errors.",
    "period_beginning": "2012-06-01 00:00:00",
    "period_ending": "2012-07-01 00:00:00",
    "total_errors": 0,
    "total_instances": 0
  }
}
```



```
}
```

4.40. Migrations (os-migrations)

Shows data on migrations.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-migrations	Lists in-progress migrations.

4.40.1. List migrations

Method	URI	Description
GET	/v2.1/{tenant_id}/os-migrations	Lists in-progress migrations.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.40.1.1. Request

This table shows the URI parameters for the list migrations request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.40.1.2. Response

Example 4.204. List migrations: JSON response

```
{
  "migrations": [
    {
      "created_at": "2012-10-29T13:42:02.000000",
      "dest_compute": "compute2",
      "dest_host": "192.0.2.0",
      "dest_node": "node2",
      "id": 1234,
      "instance_uuid": "instance_id_123",
      "new_instance_type_id": 2,
      "old_instance_type_id": 1,
      "source_compute": "compute1",
      "source_node": "node1",
      "status": "Done",
      "updated_at": "2012-10-29T13:42:02.000000"
    },
    {
      "created_at": "2013-10-22T13:42:02.000000",
      "dest_compute": "compute20",
      "dest_host": "5.6.7.8",
      "dest_node": "node20",
      "id": 5678,
      "instance_uuid": "instance_id_456",
      "new_instance_type_id": 6,
      "old_instance_type_id": 5,
      "source_compute": "compute10",
      "source_node": "node10",
      "status": "Done",
      "updated_at": "2013-10-22T13:42:02.000000"
    }
  ]
}
```

}

4.41. Networks (os-networks)

Creates, lists, shows information for, and deletes networks.

Adds network to a project, disassociates a network from a project, and disassociates a project from a network.

Associates host with and disassociates host from a network.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks	Creates a network.
GET	/v2.1/{tenant_id}/os-networks	Lists networks for the project.
POST	/v2.1/{tenant_id}/os-networks/add	Adds a network to a project.
GET	/v2.1/{tenant_id}/os-networks/{network_id}	Shows details for a network.
DELETE	/v2.1/{tenant_id}/os-networks/{network_id}	Deletes a network.
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Associates a network with a host.
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Disassociates a host from a network.
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Disassociates a network from a project. You can then reuse the network.
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Disassociates a project from a network.

4.41.1. Create network

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks	Creates a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.41.1.1. Request

This table shows the URI parameters for the create network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.205. Create network: JSON request

```
{
  "network": {
    "label": "new net 111",
    "cidr": "10.20.105.0/24",
    "mtu": 9000,
    "dhcp_server": "10.20.105.2",
    "enable_dhcp": false,
    "share_address": true,
    "allowed_start": "10.20.105.10",
    "allowed_end": "10.20.105.200"
  }
}
```

4.41.1.2. Response

Example 4.206. Create network: JSON response

```
{
  "network": {
    "bridge": null,
    "bridge_interface": null,
    "broadcast": "10.20.105.255",
    "cidr": "10.20.105.0/24",
    "cidr_v6": null,
    "created_at": null,
    "deleted": null,
    "deleted_at": null,
    "dhcp_server": "10.20.105.2",
    "dhcp_start": "10.20.105.2",
    "dns1": null,
    "dns2": null,
    "enable_dhcp": false,
    "gateway": "10.20.105.1",
    "gateway_v6": null,
  }
}
```

```
    "host": null,  
    "id": "d7a17c0c-457e-4ab4-a99c-4fa1762f5359",  
    "injected": null,  
    "label": "new net 111",  
    "mtu": 9000,  
    "multi_host": null,  
    "netmask": "255.255.255.0",  
    "netmask_v6": null,  
    "priority": null,  
    "project_id": null,  
    "rxtx_base": null,  
    "share_address": true,  
    "updated_at": null,  
    "vlan": null,  
    "vpn_private_address": null,  
    "vpn_public_address": null,  
    "vpn_public_port": null  
  }  
}
```

4.41.2. List networks

Method	URI	Description
GET	/v2.1/{tenant_id}/os-networks	Lists networks for the project.

Policy defaults enable all users to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.41.2.1. Request

This table shows the URI parameters for the list networks request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.41.2.2. Response

Example 4.207. List networks: JSON response

```
{
  "networks": [
    {
      "bridge": "br100",
      "bridge_interface": "eth0",
      "broadcast": "10.0.0.7",
      "cidr": "10.0.0.0/29",
      "cidr_v6": null,
      "created_at": "2011-08-15T06:19:19.387525",
      "deleted": false,
      "deleted_at": null,
      "dhcp_server": "10.0.0.1",
      "dhcp_start": "10.0.0.3",
      "dns1": null,
      "dns2": null,
      "enable_dhcp": true,
      "gateway": "10.0.0.1",
      "gateway_v6": null,
      "host": "nsokolov-desktop",
      "id": "20c8acc0-f747-4d71-a389-46d078ebf047",
      "injected": false,
      "label": "mynet_0",
      "mtu": null,
      "multi_host": false,
      "netmask": "255.255.255.248",
      "netmask_v6": null,
      "priority": null,
      "project_id": "1234",
      "rxtx_base": null,
      "share_address": false,
      "updated_at": "2011-08-16T09:26:13.048257",
      "vlan": 100,
    }
  ]
}
```

```
        "vpn_private_address": "10.0.0.2",
        "vpn_public_address": "127.0.0.1",
        "vpn_public_port": 1000
    },
    {
        "bridge": "br101",
        "bridge_interface": "eth0",
        "broadcast": "10.0.0.15",
        "cidr": "10.0.0.10/29",
        "cidr_v6": null,
        "created_at": "2011-08-15T06:19:19.885495",
        "deleted": false,
        "deleted_at": null,
        "dhcp_server": "10.0.0.9",
        "dhcp_start": "10.0.0.11",
        "dns1": null,
        "dns2": null,
        "enable_dhcp": true,
        "gateway": "10.0.0.9",
        "gateway_v6": null,
        "host": null,
        "id": "20c8acc0-f747-4d71-a389-46d078ebf000",
        "injected": false,
        "label": "mynet_1",
        "mtu": null,
        "multi_host": false,
        "netmask": "255.255.255.248",
        "netmask_v6": null,
        "priority": null,
        "project_id": null,
        "rxtx_base": null,
        "share_address": false,
        "updated_at": null,
        "vlan": 101,
        "vpn_private_address": "10.0.0.10",
        "vpn_public_address": null,
        "vpn_public_port": 1001
    }
]
}
```

4.41.3. Add network

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks/add	Adds a network to a project.

Policy defaults enable only users with the administrative role to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.41.3.1. Request

This table shows the URI parameters for the add network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.208. Add network: JSON request

```
{
  "id": "1"
}
```


4.41.4. Show network details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-net-works/{network_id}	Shows details for a network.

Policy defaults enable all users to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.41.4.1. Request

This table shows the URI parameters for the show network details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

This operation does not accept a request body.

4.41.4.2. Response

Example 4.209. Show network details: JSON response

```
{
  "network": {
    "bridge": "br100",
    "bridge_interface": "eth0",
    "broadcast": "10.0.0.7",
    "cidr": "10.0.0.0/29",
    "cidr_v6": null,
    "created_at": "2011-08-15T06:19:19.387525",
    "deleted": false,
    "deleted_at": null,
    "dhcp_server": "10.0.0.1",
    "dhcp_start": "10.0.0.3",
    "dns1": null,
    "dns2": null,
    "enable_dhcp": true,
    "gateway": "10.0.0.1",
    "gateway_v6": null,
    "host": "nsokolov-desktop",
    "id": "20c8acc0-f747-4d71-a389-46d078ebf047",
    "injected": false,
    "label": "mynet_0",
    "mtu": null,
    "multi_host": false,
    "netmask": "255.255.255.248",
    "netmask_v6": null,
    "priority": null,
    "project_id": "1234",
    "rxtx_base": null,
    "share_address": false,
    "updated_at": "2011-08-16T09:26:13.048257",
```

```
        "vlan": 100,  
        "vpn_private_address": "10.0.0.2",  
        "vpn_public_address": "127.0.0.1",  
        "vpn_public_port": 1000  
    }  
}
```

4.41.5. Delete network

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-net-works/{network_id}	Deletes a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.41.5.1. Request

This table shows the URI parameters for the delete network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

This operation does not accept a request body.

4.41.6. Associate host

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Associates a network with a host.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.41.6.1. Request

This table shows the URI parameters for the associate host request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

Example 4.210. Associate host: JSON request

```
{
  "associate_host": "testHost"
}
```

4.41.7. Disassociate host

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Disassociates a host from a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.41.7.1. Request

This table shows the URI parameters for the disassociate host request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

Example 4.211. Disassociate host: JSON request

```
{
  "disassociate_host": null
}
```

4.41.8. Disassociate network

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Disassociates a network from a project. You can then reuse the network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.41.8.1. Request

This table shows the URI parameters for the disassociate network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

Example 4.212. Disassociate network: JSON request

```
{
  "disassociate": null
}
```

4.41.9. Disassociate project

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks/{network_id}/action	Disassociates a project from a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.41.9.1. Request

This table shows the URI parameters for the disassociate project request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

Example 4.213. Disassociate project: JSON request

```
{
  "disassociate_project": null
}
```

4.42. Quota class (os-quota-class-sets)

Provides quota classes management support.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-quota-class-sets/{class_id}	Shows the quota for a class.
PUT	/v2.1/{tenant_id}/os-quota-class-sets/{class_id}	Updates quota for a class.

4.42.1. Show quota

Method	URI	Description
GET	/v2.1/{tenant_id}/os-quota-class-sets/{class_id}	Shows the quota for a class.

Normal response codes: 200

4.42.1.1. Request

This table shows the URI parameters for the show quota request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{class_id}	UUID	The UUID of the quota class.

This operation does not accept a request body.

4.42.1.2. Response

Example 4.214. Show quota: JSON response

```
{
  "quota_class_set": {
    "cores": 20,
    "fixed_ips": -1,
    "floating_ips": 10,
    "id": "test_class",
    "injected_file_content_bytes": 10240,
    "injected_file_path_bytes": 255,
    "injected_files": 5,
    "instances": 10,
    "key_pairs": 100,
    "metadata_items": 128,
    "ram": 51200,
    "security_group_rules": 20,
    "security_groups": 10
  }
}
```


4.42.2. Update quota

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-quota-class-sets/{class_id}	Updates quota for a class.

Normal response codes: 200

4.42.2.1. Request

This table shows the URI parameters for the update quota request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{class_id}	UUID	The UUID of the quota class.

Example 4.215. Update quota: JSON request

```
{
  "quota_class_set": {
    "instances": 50,
    "cores": 50,
    "ram": 51200,
    "floating_ips": 10,
    "metadata_items": 128,
    "injected_files": 5,
    "injected_file_content_bytes": 10240,
    "injected_file_path_bytes": 255,
    "security_groups": 10,
    "security_group_rules": 20,
    "key_pairs": 100
  }
}
```

4.42.2.2. Response

Example 4.216. Update quota: JSON response

```
{
  "quota_class_set": {
    "cores": 50,
    "fixed_ips": -1,
    "floating_ips": 10,
    "injected_file_content_bytes": 10240,
    "injected_file_path_bytes": 255,
    "injected_files": 5,
    "instances": 50,
    "key_pairs": 100,
    "metadata_items": 128,
    "ram": 51200,
    "security_group_rules": 20,
    "security_groups": 10
  }
}
```

4.43. Quota sets (os-quota-sets)

Permits administrators, depending on policy settings, to view default quotas, view details for quotas, revert quotas to defaults, and update the quotas for a project or a project and user.

Method	URI	Description
PUT	/v2.1/{admin_tenant_id}/os-quota-sets/{tenant_id}{?user_id}	Update the quotas for a project or a project and a user.
DELETE	/v2.1/{admin_tenant_id}/os-quota-sets/{tenant_id}{?user_id}	Reverts the quotas to default values for a project or a project and a user.
GET	/v2.1/{admin_tenant_id}/os-quota-sets/{tenant_id}/defaults	Lists the default quotas for a project.
GET	/v2.1/{admin_tenant_id}/os-quota-sets/{tenant_id}/detail	Lists quotas with details for a project or a project and a user.

4.43.1. Update quotas

Method	URI	Description
PUT	/v2.1/{admin_tenant_id}/os-quota-sets/{tenant_id}{?user_id}	Update the quotas for a project or a project and a user.

You can force the update even if the quota has already been used and the reserved quota exceeds the new quota.

To force the update, specify the "force": "True" attribute in the request body. Default is false. The following example request shows the optional force attribute.

To update a quota for a project and a user, specify the user_id query parameter.

Normal response codes: 200

4.43.1.1. Request

This table shows the URI parameters for the update quotas request:

Name	Type	Description
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the update quotas request:

Name	Type	Description
user_id	String (Optional)	Filters the response by a user, by ID.

Example 4.217. Update quotas: JSON request

```
{
  "quota_set": {
    "force": true,
    "key_pairs": 120
  }
}
```

4.43.1.2. Response

Example 4.218. Update quotas: JSON response

```
{
  "quota_set": {
    "injected_file_content_bytes": 10240,
    "metadata_items": 128,
    "server_group_members": 10,
    "server_groups": 10,
    "ram": 51200,
    "floating_ips": 10,
    "key_pairs": 120,
    "instances": 10,
  }
}
```

```
    "security_group_rules": 20,  
    "injected_files": 5,  
    "cores": 20,  
    "fixed_ips": -1,  
    "injected_file_path_bytes": 255,  
    "security_groups": 10  
  }  
}
```

4.43.2. Revert quotas to defaults

Method	URI	Description
DELETE	/v2.1/{admin_tenant_id}/os-quotas-sets/{tenant_id}{?user_id}	Reverts the quotas to default values for a project or a project and a user.

To revert quotas for a project and a user, specify the `user_id` query parameter.

Normal response codes: 202

4.43.2.1. Request

This table shows the URI parameters for the revert quotas to defaults request:

Name	Type	Description
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the revert quotas to defaults request:

Name	Type	Description
user_id	String (Optional)	Filters the response by a user, by ID.

4.43.3. List default quotas for tenant

Method	URI	Description
GET	/v2.1/{admin_tenant_id}/os-quotas/{tenant_id}/defaults	Lists the default quotas for a project.

In the request URI, you specify both the ID of the administrative project and the ID of the project for which you want to show default quotas. These project IDs can match.

Normal response codes: 200

4.43.3.1. Request

This table shows the URI parameters for the list default quotas for tenant request:

Name	Type	Description
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.43.3.2. Response

Example 4.219. List default quotas for tenant: JSON response

```
{
  "quota_set": {
    "injected_file_content_bytes": 10240,
    "metadata_items": 128,
    "server_group_members": 10,
    "server_groups": 10,
    "ram": 51200,
    "floating_ips": 10,
    "key_pairs": 100,
    "id": "91a3c6da787643c78f2a7c7428fa54f2",
    "instances": 10,
    "security_group_rules": 20,
    "injected_files": 5,
    "cores": 20,
    "fixed_ips": -1,
    "injected_file_path_bytes": 255,
    "security_groups": 10
  }
}
```

4.43.4. List quotas with details

Method	URI	Description
GET	/v2.1/{admin_tenant_id}/os-quotas/{tenant_id}/detail	Lists quotas with details for a project or a project and a user.

To list a quota for a project and a user, specify the `user_id` query parameter.

Normal response codes: 200

4.43.4.1. Request

This table shows the URI parameters for the list quotas with details request:

Name	Type	Description
{admin_tenant_id}	UUID	The UUID of the administrative tenant.
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.43.4.2. Response

Example 4.220. List quotas with details: JSON response

```
{
  "quota_set": {
    "injected_file_content_bytes": {
      "reserved": 0,
      "limit": 10240,
      "in_use": 0
    },
    "metadata_items": {
      "reserved": 0,
      "limit": 128,
      "in_use": 0
    },
    "server_group_members": {
      "reserved": 0,
      "limit": 10,
      "in_use": 0
    },
    "server_groups": {
      "reserved": 0,
      "limit": 10,
      "in_use": 0
    },
    "ram": {
      "reserved": 0,
      "limit": 51200,
      "in_use": 0
    },
    "floating_ips": {
      "reserved": 0,
      "limit": 10,
      "in_use": 0
    }
  }
}
```

```

    },
    "key_pairs": {
        "reserved": 0,
        "limit": 120,
        "in_use": 0
    },
    "id": "91a3c6da787643c78f2a7c7428fa54f2",
    "instances": {
        "reserved": 0,
        "limit": 10,
        "in_use": 0
    },
    "security_group_rules": {
        "reserved": 0,
        "limit": 20,
        "in_use": 0
    },
    "injected_files": {
        "reserved": 0,
        "limit": 5,
        "in_use": 0
    },
    "cores": {
        "reserved": 0,
        "limit": 20,
        "in_use": 0
    },
    "fixed_ips": {
        "reserved": 0,
        "limit": -1,
        "in_use": 0
    },
    "injected_file_path_bytes": {
        "reserved": 0,
        "limit": 255,
        "in_use": 0
    },
    "security_groups": {
        "reserved": 0,
        "limit": 10,
        "in_use": 0
    }
}

```

4.44. Security groups (os-security-groups)

Lists, shows information for, creates, and deletes security groups.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-security-groups	Lists security groups.
POST	/v2.1/{tenant_id}/os-security-groups	Creates a security group.
GET	/v2.1/{tenant_id}/os-security-groups/{security_group_id}	Shows details for a security group.
PUT	/v2.1/{tenant_id}/os-security-groups/{security_group_id}	Updates a security group.

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-security-groups/{security_group_id}	Deletes a security group.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-security-groups	Lists security groups for a server.

4.44.1. List security groups

Method	URI	Description
GET	/v2.1/{tenant_id}/os-security-groups	Lists security groups.

Normal response codes: 200

4.44.1.1. Request

This table shows the URI parameters for the list security groups request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.44.1.2. Response

Example 4.221. List security groups: JSON response

```
{
  "security_groups": [
    {
      "description": "default",
      "id": 1,
      "name": "default",
      "rules": [],
      "tenant_id": "openstack"
    }
  ]
}
```

4.44.2. Create security group

Method	URI	Description
POST	/v2.1/{tenant_id}/os-security-groups	Creates a security group.

Normal response codes: 200

4.44.2.1. Request

This table shows the URI parameters for the create security group request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.222. Create security group: JSON request

```
{
  "security_group": {
    "name": "test",
    "description": "description"
  }
}
```

4.44.2.2. Response

Example 4.223. Create security group: JSON response

```
{
  "security_group": {
    "description": "default",
    "id": 1,
    "name": "default",
    "rules": [],
    "tenant_id": "openstack"
  }
}
```

4.44.3. Show security group details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-security-groups/{security_group_id}	Shows details for a security group.

Normal response codes: 200

4.44.3.1. Request

This table shows the URI parameters for the show security group details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{security_group_id}	Int	The ID of the security group.

This operation does not accept a request body.

4.44.3.2. Response

Example 4.224. Show security group: JSON response

```
{
  "security_group": {
    "description": "default",
    "id": 1,
    "name": "default",
    "rules": [],
    "tenant_id": "openstack"
  }
}
```

4.44.4. Update security group

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-security-groups/{security_group_id}	Updates a security group.

Normal response codes: 200

4.44.4.1. Request

This table shows the URI parameters for the update security group request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{security_group_id}	Int	The ID of the security group.

Example 4.225. Update security group: JSON request

```
{
  "security_group": {
    "name": "mysecgroup",
    "description": "my security group"
  }
}
```

4.44.4.2. Response

Example 4.226. Update security group: JSON response

```
{
  "security_group": {
    "description": "my security group",
    "id": 1,
    "name": "mysecgroup",
    "rules": [],
    "tenant_id": "a52cdb9cc7854a39a23d3af73a40899e"
  }
}
```

4.44.5. Delete security group

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-security-groups/{security_group_id}	Deletes a security group.

Normal response codes: 202

4.44.5.1. Request

This table shows the URI parameters for the delete security group request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{security_group_id}	Int	The ID of the security group.

This operation does not accept a request body.

4.44.6. List security groups by server

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-security-groups	Lists security groups for a server.

Normal response codes: 200

4.44.6.1. Request

This table shows the URI parameters for the list security groups by server request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server.

This operation does not accept a request body.

4.44.6.2. Response

Example 4.227. List security groups by server: JSON response

```
{
  "security_groups": [
    {
      "description": "default",
      "id": 1,
      "name": "default",
      "rules": [],
      "tenant_id": "openstack"
    }
  ]
}
```

4.45. Rules for default security group (os-security-group-default-rules)

Lists, shows information for, and creates default security group rules.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-security-group-default-rules	Lists default security group rules.
POST	/v2.1/{tenant_id}/os-security-group-default-rules	Creates a default security group rule.
GET	/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Shows details for a security group rule.
DELETE	/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Deletes a security group rule.

4.45.1. List default security group rules

Method	URI	Description
GET	/v2.1/{tenant_id}/os-security-group-default-rules	Lists default security group rules.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.45.1.1. Request

This table shows the URI parameters for the list default security group rules request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.45.1.2. Response

Example 4.228. List default security group rules: JSON response

```
{
  "security_group_default_rules": [
    {
      "from_port": 80,
      "id": 1,
      "ip_protocol": "TCP",
      "ip_range": {
        "cidr": "10.10.10.0/24"
      },
      "to_port": 80
    }
  ]
}
```


4.45.2. Create default security group rule

Method	URI	Description
POST	/v2.1/{tenant_id}/os-security-group-default-rules	Creates a default security group rule.

If you specify a source port (`from_port`) or destination port (`to_port`) value, you must specify an IP protocol (`ip_protocol`) value. Otherwise, the operation returns the Bad Request (400) response code.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.45.2.1. Request

This table shows the URI parameters for the create default security group rule request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.229. Create default security group rule: JSON request

```
{
  "security_group_default_rule": {
    "ip_protocol": "TCP",
    "from_port": "80",
    "to_port": "80",
    "cidr": "10.10.10.0/24"
  }
}
```

4.45.2.2. Response

Example 4.230. Create default security group rule: JSON response

```
{
  "security_group_default_rule": {
    "from_port": 80,
    "id": 1,
    "ip_protocol": "TCP",
    "ip_range": {
      "cidr": "10.10.10.0/24"
    },
    "to_port": 80
  }
}
```

4.45.3. Show default security group rule details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Shows details for a security group rule.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.45.3.1. Request

This table shows the URI parameters for the show default security group rule details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{security_group_default_rule_id}	UUID	The UUID of the security group rule.

This operation does not accept a request body.

4.45.3.2. Response

Example 4.231. Show default security group rule: JSON response

```
{
  "security_group_default_rule": {
    "from_port": 80,
    "id": 1,
    "ip_protocol": "TCP",
    "ip_range": {
      "cidr": "10.10.10.0/24"
    },
    "to_port": 80
  }
}
```

4.45.4. Delete default security group rule

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Deletes a security group rule.

Normal response codes: 204

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.45.4.1. Request

This table shows the URI parameters for the delete default security group rule request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{security_group_default_rule_id}	UUID	The UUID of the security group rule.

This operation does not accept a request body.

4.46. Rules for security group (os-security-group-rules)

Creates and deletes security group rules.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-security-group-rules	Creates a rule for a security group.
DELETE	/v2.1/{tenant_id}/os-security-group-rules/{security_group_rule_id}	Deletes a security group rule.

4.46.1. Create security group rule

Method	URI	Description
POST	/v2.1/{tenant_id}/os-security-group-rules	Creates a rule for a security group.

Normal response codes: 200

4.46.1.1. Request

This table shows the URI parameters for the create security group rule request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.232. Create security group rule: JSON request

```
{
  "security_group_rule": {
    "from_port": "443",
    "ip_protocol": "tcp",
    "to_port": "443",
    "cidr": "0.0.0.0/0",
    "parent_group_id": "48700ff3-30b8-4e63-845f-a79c9633e9fb"
  }
}
```

4.46.1.2. Response

Example 4.233. Create security group rule: JSON response

```
{
  "security_group_rule": {
    "id": "1",
    "ip_range": {
      "cidr": "0.0.0.0/0"
    },
    "parent_group_id": "48700ff3-30b8-4e63-845f-a79c9633e9fb",
    "to_port": 443,
    "ip_protocol": "tcp",
    "group": {},
    "from_port": 443
  }
}
```

4.46.2. Delete security group rule

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-security-group-rules/{security_group_rule_id}	Deletes a security group rule.

Normal response codes: 202

4.46.2.1. Request

This table shows the URI parameters for the delete security group rule request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{security_group_rule_id}	Int	The ID of the security group rule.

This operation does not accept a request body.

4.47. Create external events (os-server-external-events)

Creates one or more external events. The API dispatches each event to a server instance.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-server-external-events	Creates one or more external events, which the API dispatches to the instance.

4.47.1. Run events

Method	URI	Description
POST	/v2.1/{tenant_id}/os-server-external-events	Creates one or more external events, which the API dispatches to the instance.

You must assign this instance to a host. Otherwise, this call does not dispatch the event to the instance.

Normal response codes: 200

4.47.1.1. Request

This table shows the URI parameters for the run events request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.234. Run events: JSON request

```
{
  "events": [
    {
      "name": "test-event",
      "tag": "foo",
      "status": "completed",
      "server_uuid": "3df201cf-2451-44f2-8d25-a4ca826fc1f3"
    }
  ]
}
```

4.47.1.2. Response

Example 4.235. Run events: JSON response

```
{
  "events": [
    {
      "code": 200,
      "name": "network-changed",
      "server_uuid": "ff1df7b2-6772-45fd-9326-c0a3b05591c2",
      "status": "completed",
      "tag": "foo"
    }
  ]
}
```

4.48. Server groups (os-server-groups)

Lists, shows information for, creates, and deletes server groups.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-server-groups {?all_projects}	Lists all server groups for the tenant.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-server-groups	Creates a server group.
GET	/v2.1/{tenant_id}/os-server-groups/{server_group_id}	Shows details for a server group.
DELETE	/v2.1/{tenant_id}/os-server-groups/{server_group_id}	Deletes a server group.

4.48.1. List server groups

Method	URI	Description
GET	/v2.1/{tenant_id}/os-server-groups {?all_projects}	Lists all server groups for the tenant.

Administrative users can use the `all_projects` query parameter to list all server groups for all projects.

Normal response codes: 200

4.48.1.1. Request

This table shows the URI parameters for the list server groups request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list server groups request:

Name	Type	Description
all_projects	Boolean <i>(Optional)</i>	Administrator only. Lists server groups for all projects. For example: <code>GET /v2.1/{admin_tenant_id}/os-server-groups?all_projects=true</code> If you specify a tenant ID for a non-administrative user with this query parameter, the call lists all server groups for the tenant, or project, rather than for all projects.

This operation does not accept a request body.

4.48.1.2. Response

Example 4.236. List server groups: JSON response

```
{
  "server_groups": [
    {
      "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
      "name": "test",
      "policies": [
        "anti-affinity"
      ],
      "members": [],
      "metadata": {},
      "project_id": "test-project",
      "user_id": "test-user"
    }
  ]
}
```


4.48.2. Create server group

Method	URI	Description
POST	/v2.1/{tenant_id}/os-server-groups	Creates a server group.

Normal response codes: 200

4.48.2.1. Request

This table shows the URI parameters for the create server group request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.237. Create server group: JSON request

```
{
  "server_group": {
    "name": "test",
    "policies": [
      "anti-affinity"
    ]
  }
}
```

4.48.2.2. Response

Example 4.238. Create server group: JSON response

```
{
  "server_group": {
    "id": "5bbcc3c4-1da2-4437-a48a-66f15b1b13f9",
    "name": "test",
    "policies": [
      "anti-affinity"
    ],
    "members": [],
    "metadata": {},
    "project_id": "test-project",
    "user_id": "test-user"
  }
}
```

4.48.3. Show server group details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-server-groups/{server_group_id}	Shows details for a server group.

Normal response codes: 200

4.48.3.1. Request

This table shows the URI parameters for the show server group details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_group_id}	UUID	The UUID of the server group.

This operation does not accept a request body.

4.48.3.2. Response

Example 4.239. Show server group details: JSON response

```
{
  "server_group": {
    "id": "5bbcc3c4-1da2-4437-a48a-66f15b1b13f9",
    "name": "test",
    "policies": [
      "anti-affinity"
    ],
    "members": [],
    "metadata": {},
    "project_id": "test-project",
    "user_id": "test-user"
  }
}
```

4.48.4. Delete server group

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-server-groups/{server_group_id}	Deletes a server group.

Normal response codes: 204

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.48.4.1. Request

This table shows the URI parameters for the delete server group request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{server_group_id}	UUID	The UUID of the server group.

This operation does not accept a request body.

4.49. Compute services (os-services)

Lists all running Compute services for a tenant, enables or disables scheduling for a Compute service, logs disabled Compute service information, and deletes a Compute service.

For an overview of Compute services, see [OpenStack Compute](#).

Method	URI	Description
GET	/v2.1/{tenant_id}/os-services	Lists all running Compute services for a tenant.
PUT	/v2.1/{tenant_id}/os-services/enable	Enables scheduling for a Compute service.
PUT	/v2.1/{tenant_id}/os-services/disable	Disables scheduling for a Compute service.
PUT	/v2.1/{tenant_id}/os-services/disable-log-reason	Logs information to the Compute service table about why a Compute service was disabled.
DELETE	/v2.1/{tenant_id}/os-services/{service_id}	Deletes a Compute service.

4.49.1. List Compute services

Method	URI	Description
GET	/v2.1/{tenant_id}/os-services	Lists all running Compute services for a tenant.

Includes reasons, if available, for why any services were disabled.

Normal response codes: 200

4.49.1.1. Request

This table shows the URI parameters for the list compute services request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.49.1.2. Response

Example 4.240. List Compute services: JSON response

```
{
  "services": [
    {
      "id": 1,
      "binary": "nova-scheduler",
      "disabled_reason": "test1",
      "host": "host1",
      "state": "up",
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:02.000000",
      "forced_down": false,
      "zone": "internal"
    },
    {
      "id": 2,
      "binary": "nova-compute",
      "disabled_reason": "test2",
      "host": "host1",
      "state": "up",
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:05.000000",
      "forced_down": false,
      "zone": "nova"
    },
    {
      "id": 3,
      "binary": "nova-scheduler",
      "disabled_reason": null,
      "host": "host2",
      "state": "down",
      "status": "enabled",
      "updated_at": "2012-09-19T06:55:34.000000",
      "forced_down": false,
    }
  ]
}
```

```
        "zone": "internal"
    },
    {
        "id": 4,
        "binary": "nova-compute",
        "disabled_reason": "test4",
        "host": "host2",
        "state": "down",
        "status": "disabled",
        "updated_at": "2012-09-18T08:03:38.000000",
        "forced_down": false,
        "zone": "nova"
    }
]
}
```

4.49.2. Enable scheduling for a Compute service

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-services/en-able	Enables scheduling for a Compute service.

Specify the service by its host name and binary name.

Normal response codes: 200

4.49.2.1. Request

This table shows the URI parameters for the enable scheduling for a compute service request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.241. Enable scheduling for a Compute service: JSON request

```
{
  "host": "host1",
  "binary": "nova-compute"
}
```

4.49.2.2. Response

Example 4.242. Enable scheduling for a Compute service: JSON response

```
{
  "service": {
    "binary": "nova-compute",
    "host": "host1",
    "status": "enabled"
  }
}
```

4.49.3. Disable scheduling for a Compute service

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-services/disable	Disables scheduling for a Compute service.

Specify the service by its host name and binary name.

Normal response codes: 200

4.49.3.1. Request

This table shows the URI parameters for the disable scheduling for a compute service request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.243. Disable scheduling for a Compute service: JSON request

```
{
  "host": "host1",
  "binary": "nova-compute"
}
```

4.49.3.2. Response

Example 4.244. Disable scheduling for a Compute service: JSON response

```
{
  "service": {
    "binary": "nova-compute",
    "host": "host1",
    "status": "disabled"
  }
}
```

4.49.4. Log disabled Compute service information

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-services/disable-log-reason	Logs information to the Compute service table about why a Compute service was disabled.

Specify the service by its host name and binary name.

Normal response codes: 200

4.49.4.1. Request

This table shows the URI parameters for the log disabled compute service information request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.245. Log disabled Compute service information: JSON request

```
{
  "host": "host1",
  "binary": "nova-compute",
  "disabled_reason": "test2"
}
```

4.49.4.2. Response

Example 4.246. Log disabled Compute service information: JSON response

```
{
  "service": {
    "binary": "nova-compute",
    "disabled_reason": "test2",
    "host": "host1",
    "status": "disabled"
  }
}
```


4.49.5. Delete Compute service

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-services/{service_id}	Deletes a Compute service.

Specify the service by its host name and binary name.

Normal response codes: 204

Error response codes: 404

4.49.5.1. Request

This table shows the URI parameters for the delete compute service request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{service_id}	UUID	The UUID of the service.

Example 4.247. Delete Compute service: JSON request

```
{
  "host": "host1",
  "binary": "nova-compute"
}
```

4.50. Usage reports (os-simple-tenant-usage)

Reports usage statistics on compute and storage resources.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-simple-tenant-usage	Lists usage information for all tenants.
GET	/v2.1/{tenant_id}/os-simple-tenant-usage/{tenant_id}	Shows usage details for a tenant.

4.50.1. List tenant usage for all tenants

Method	URI	Description
GET	/v2.1/{tenant_id}/os-simple-tenant-usage	Lists usage information for all tenants.

Normal response codes: 200

4.50.1.1. Request

This table shows the URI parameters for the list tenant usage for all tenants request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

4.50.1.2. Response

Example 4.248. List tenant usage for all tenants: JSON response

```
{
  "tenant_usages": [
    {
      "start": "2012-10-08T21:10:44.587336",
      "stop": "2012-10-08T22:10:44.587336",
      "tenant_id": "openstack",
      "total_hours": 1.0,
      "total_local_gb_usage": 1.0,
      "total_memory_mb_usage": 512.0,
      "total_vcpus_usage": 1.0
    }
  ]
}
```

4.50.2. Show usage details for tenant

Method	URI	Description
GET	/v2.1/{tenant_id}/os-simple-tenant-usage/{tenant_id}	Shows usage details for a tenant.

Normal response codes: 200

4.50.2.1. Request

This table shows the URI parameters for the show usage details for tenant request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

4.50.2.2. Response

Example 4.249. Show usage details for tenant: JSON response

```
{
  "tenant_usage": {
    "server_usages": [
      {
        "ended_at": null,
        "flavor": "m1.tiny",
        "hours": 1.0,
        "instance_id": "1f1deceb-17b5-4c04-84c7-e0d4499c8fe0",
        "local_gb": 1,
        "memory_mb": 512,
        "name": "new-server-test",
        "started_at": "2012-10-08T20:10:44.541277",
        "state": "active",
        "tenant_id": "openstack",
        "uptime": 3600,
        "vcpus": 1
      }
    ],
    "start": "2012-10-08T20:10:44.587336",
    "stop": "2012-10-08T21:10:44.587336",
    "tenant_id": "openstack",
    "total_hours": 1.0,
    "total_local_gb_usage": 1.0,
    "total_memory_mb_usage": 512.0,
    "total_vcpus_usage": 1.0
  }
}
```

4.51. Project networks (os-tenant-networks)

Creates, lists, shows information for, and deletes project networks.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-tenant-networks	Creates a project network.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-tenant-networks	Lists all project networks.
GET	/v2.1/{tenant_id}/os-tenant-networks/{network_id}	Shows details for a project network.
DELETE	/v2.1/{tenant_id}/os-tenant-networks/{network_id}	Deletes a project network.
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Associates a network with a host.
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Disassociates a host from a network.
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Disassociates a network from a project so that the network can be reused.
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Disassociates a project from a network.

4.51.1. Create project network

Method	URI	Description
POST	/v2.1/{tenant_id}/os-tenant-networks	Creates a project network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.51.1.1. Request

This table shows the URI parameters for the create project network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.250. Create project network: JSON request

```
{
  "network": {
    "label": "public",
    "cidr": "172.0.0.0/24",
    "vlan_start": 1,
    "num_networks": 1,
    "network_size": 255
  }
}
```

4.51.1.2. Response

Example 4.251. Create project network: JSON response

```
{
  "network": {
    "cidr": "172.0.0.0/24",
    "id": "5bbcc3c4-1da2-4437-a48a-66f15b1b13f9",
    "label": "public"
  }
}
```

4.51.2. List project networks

Method	URI	Description
GET	/v2.1/{tenant_id}/os-tenant-networks	Lists all project networks.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.51.2.1. Request

This table shows the URI parameters for the list project networks request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.51.2.2. Response

Example 4.252. List project networks: JSON response

```
{
  "networks": [
    {
      "bridge": "br100",
      "bridge_interface": "eth0",
      "broadcast": "10.0.0.7",
      "cidr": "10.0.0.0/29",
      "cidr_v6": null,
      "created_at": "2011-08-15T06:19:19.387525",
      "deleted": false,
      "deleted_at": null,
      "dhcp_server": "10.0.0.1",
      "dhcp_start": "10.0.0.3",
      "dns1": null,
      "dns2": null,
      "enable_dhcp": true,
      "gateway": "10.0.0.1",
      "gateway_v6": null,
      "host": "nsokolov-desktop",
      "id": "20c8acc0-f747-4d71-a389-46d078ebf047",
      "injected": false,
      "label": "mynet_0",
      "mtu": null,
      "multi_host": false,
      "netmask": "255.255.255.248",
      "netmask_v6": null,
      "priority": null,
      "project_id": "1234",
      "rxtx_base": null,
      "share_address": false,
    }
  ]
}
```

```
        "updated_at": "2011-08-16T09:26:13.048257",
        "vlan": 100,
        "vpn_private_address": "10.0.0.2",
        "vpn_public_address": "127.0.0.1",
        "vpn_public_port": 1000
    },
    {
        "bridge": "br101",
        "bridge_interface": "eth0",
        "broadcast": "10.0.0.15",
        "cidr": "10.0.0.10/29",
        "cidr_v6": null,
        "created_at": "2011-08-15T06:19:19.885495",
        "deleted": false,
        "deleted_at": null,
        "dhcp_server": "10.0.0.9",
        "dhcp_start": "10.0.0.11",
        "dns1": null,
        "dns2": null,
        "enable_dhcp": true,
        "gateway": "10.0.0.9",
        "gateway_v6": null,
        "host": null,
        "id": "20c8acc0-f747-4d71-a389-46d078ebf000",
        "injected": false,
        "label": "mynet_1",
        "mtu": null,
        "multi_host": false,
        "netmask": "255.255.255.248",
        "netmask_v6": null,
        "priority": null,
        "project_id": null,
        "rxtx_base": null,
        "share_address": false,
        "updated_at": null,
        "vlan": 101,
        "vpn_private_address": "10.0.0.10",
        "vpn_public_address": null,
        "vpn_public_port": 1001
    }
]
```

4.51.3. Show project network details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-tenant-networks/{network_id}	Shows details for a project network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

4.51.3.1. Request

This table shows the URI parameters for the show project network details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

This operation does not accept a request body.

4.51.3.2. Response

Example 4.253. Show project network details: JSON response

```
{
  "networks": [
    {
      "cidr": "10.0.0.0/29",
      "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
      "label": "test_0"
    },
    {
      "cidr": "10.0.0.8/29",
      "id": "616fb98f-46ca-475e-917e-2563e5a8cd20",
      "label": "test_1"
    }
  ]
}
```


4.51.4. Delete project network

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-tenant-networks/{network_id}	Deletes a project network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.51.4.1. Request

This table shows the URI parameters for the delete project network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

This operation does not accept a request body.

4.51.5. Associate host

Method	URI	Description
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Associates a network with a host.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.51.5.1. Request

This table shows the URI parameters for the associate host request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

Example 4.254. Associate host: JSON request

```
{
  "associate_host": "testHost"
}
```

4.51.6. Disassociate host

Method	URI	Description
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Disassociates a host from a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.51.6.1. Request

This table shows the URI parameters for the disassociate host request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

Example 4.255. Disassociate host: JSON request

```
{
  "disassociate_host": null
}
```

4.51.7. Disassociate network

Method	URI	Description
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Disassociates a network from a project so that the network can be reused.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.51.7.1. Request

This table shows the URI parameters for the disassociate network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

Example 4.256. Disassociate network: JSON request

```
{
  "disassociate": null
}
```

4.51.8. Disassociate project

Method	URI	Description
POST	/v2.1/{tenant_id}/os-tenant-networks/{network_id}/action	Disassociates a project from a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

4.51.8.1. Request

This table shows the URI parameters for the disassociate project request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{network_id}	UUID	The UUID of the network.

Example 4.257. Disassociate project: JSON request

```
{
  "disassociate_project": null
}
```

4.52. Volume extension (os-volumes, os-snapshots)

Manages volumes and snapshots for use with the Compute API.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-volumes	Lists the volumes associated with the account.
POST	/v2.1/{tenant_id}/os-volumes	Creates a volume.
GET	/v2.1/{tenant_id}/os-volumes/detail	Lists all volumes with details.
GET	/v2.1/{tenant_id}/os-volumes/{volume_id}	Shows details for a volume.
DELETE	/v2.1/{tenant_id}/os-volumes/{volume_id}	Deletes a volume.
GET	/v2.1/{tenant_id}/os-volume-types	Lists volume types.
GET	/v2.1/{tenant_id}/os-volume-types/{volume_type_id}	Shows details for a volume type.
POST	/v2.1/{tenant_id}/os-snapshots	Creates a snapshot.
GET	/v2.1/{tenant_id}/os-snapshots	Lists snapshots.
GET	/v2.1/{tenant_id}/os-snapshots/detail	Lists all snapshots with details.
GET	/v2.1/{tenant_id}/os-snapshots/{snapshot_id}	Shows details for a snapshot.

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-snapshots/{snapshot_id}	Deletes a snapshot from the account.

4.52.1. List volumes

Method	URI	Description
GET	/v2.1/{tenant_id}/os-volumes	Lists the volumes associated with the account.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.1.1. Request

This table shows the URI parameters for the list volumes request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.52.1.2. Response

Example 4.258. List volumes: JSON response

```
{
  "volumes": [
    {
      "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
      "display_name": "vol-001",
      "display_description": "Another volume.",
      "size": 30,
      "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
      "metadata": {
        "contents": "junk"
      },
      "availability_zone": "us-east1",
      "snapshot_id": null,
      "attachments": [],
      "created_at": "2012-02-14T20:53:07Z"
    },
    {
      "id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
      "display_name": "vol-002",
      "display_description": "Yet another volume.",
      "size": 25,
      "volume_type": "96c3bda7-c82a-4f50-be73-ca7621794835",
      "metadata": {},
      "availability_zone": "us-east2",
      "snapshot_id": null,
      "attachments": [],
      "created_at": "2012-03-15T19:10:03Z"
    }
  ]
}
```

4.52.2. Create volume

Method	URI	Description
POST	/v2.1/{tenant_id}/os-volumes	Creates a volume.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.2.1. Request

This table shows the URI parameters for the create volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.259. Create volume: JSON request

```
{
  "volume": {
    "display_name": "vol-001",
    "display_description": "Another volume.",
    "size": 30,
    "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
    "metadata": {
      "contents": "junk"
    },
    "availability_zone": "us-east1"
  }
}
```

4.52.2.2. Response

Example 4.260. Create volume: JSON response

```
{
  "volume": {
    "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "display_name": "vol-001",
    "display_description": "Another volume.",
    "status": "active",
    "size": 30,
    "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
    "metadata": {
      "contents": "junk"
    },
    "availability_zone": "us-east1",
    "snapshot_id": null,
    "attachments": [],
    "created_at": "2012-02-14T20:53:07Z"
  }
}
```


4.52.3. List volumes with details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-volumes/detail	Lists all volumes with details.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.3.1. Request

This table shows the URI parameters for the list volumes with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.52.3.2. Response

Example 4.261. List volumes with details: JSON response

```
{
  "volumes": [
    {
      "attachments": [
        {
          "device": "/",
          "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
          "serverId": "3912f2b4-c5ba-4aec-9165-872876fe202e",
          "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
        }
      ],
      "availabilityZone": "zone1:host1",
      "createdAt": "1999-01-01T01:01:01.000000",
      "displayDescription": "Volume Description",
      "displayName": "Volume Name",
      "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
      "metadata": {},
      "size": 100,
      "snapshotId": null,
      "status": "in-use",
      "volumeType": "Backup"
    }
  ]
}
```

4.52.4. Show volume details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-volumes/{volume_id}	Shows details for a volume.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.4.1. Request

This table shows the URI parameters for the show volume details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	String	The unique ID for a volume.

This operation does not accept a request body.

4.52.4.2. Response

Example 4.262. Show volume details: JSON response

```
{
  "volume": {
    "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "display_name": "vol-001",
    "display_description": "Another volume.",
    "status": "active",
    "size": 30,
    "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
    "metadata": {
      "contents": "junk"
    },
    "availability_zone": "us-east1",
    "snapshot_id": null,
    "attachments": [],
    "created_at": "2012-02-14T20:53:07Z"
  }
}
```

4.52.5. Delete volume

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-volumes/{volume_id}	Deletes a volume.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.5.1. Request

This table shows the URI parameters for the delete volume request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_id}	String	The unique ID for a volume.

This operation does not accept a request body.

4.52.5.2. Response

Example 4.263. Delete volume: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Date: Fri, 05 Dec 2014 00:39:32 GMT
```

This operation does not return a response body.

4.52.6. List volume types

Method	URI	Description
GET	/v2.1/{tenant_id}/os-volume-types	Lists volume types.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.6.1. Request

This table shows the URI parameters for the list volume types request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.52.6.2. Response

Example 4.264. List volume types: JSON response

```
{
  "volume_types": [
    {
      "id": "289da7f8-6440-407c-9fb4-7db01ec49164",
      "name": "vol-type-001",
      "extra_specs": {
        "capabilities": "gpu"
      }
    },
    {
      "id": "96c3bda7-c82a-4f50-be73-ca7621794835",
      "name": "vol-type-002",
      "extra_specs": {}
    }
  ]
}
```

4.52.7. Show volume type details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-volume-types/{volume_type_id}	Shows details for a volume type.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.7.1. Request

This table shows the URI parameters for the show volume type details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{volume_type_id}	String	The unique ID for a volume type.

This operation does not accept a request body.

4.52.7.2. Response

Example 4.265. Show volume type details: JSON response

```
{
  "volume_type": {
    "id": "289da7f8-6440-407c-9fb4-7db01ec49164",
    "name": "vol-type-001",
    "extra_specs": {
      "capabilities": "gpu"
    }
  }
}
```

4.52.8. Create snapshot

Method	URI	Description
POST	/v2.1/{tenant_id}/os-snapshots	Creates a snapshot.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.8.1. Request

This table shows the URI parameters for the create snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 4.266. Create snapshot: JSON request

```
{
  "snapshot": {
    "display_name": "snap-001",
    "display_description": "Daily backup",
    "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "force": false
  }
}
```

4.52.8.2. Response

Example 4.267. Create snapshot: JSON response

```
{
  "snapshot": {
    "createdAt": "2013-02-25T16:27:54.724209",
    "displayDescription": "Default description",
    "displayName": "Default name",
    "id": "100",
    "size": 100,
    "status": "available",
    "volumeId": 12
  }
}
```

4.52.9. List snapshots

Method	URI	Description
GET	/v2.1/{tenant_id}/os-snapshots	Lists snapshots.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.9.1. Request

This table shows the URI parameters for the list snapshots request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.52.9.2. Response

Example 4.268. List snapshots: JSON response

```
{
  "snapshots": [
    {
      "createdAt": "2013-02-25T16:27:54.684999",
      "displayDescription": "Default description",
      "displayName": "Default name",
      "id": 100,
      "size": 100,
      "status": "available",
      "volumeId": 12
    },
    {
      "createdAt": "2013-02-25T16:27:54.685005",
      "displayDescription": "Default description",
      "displayName": "Default name",
      "id": 101,
      "size": 100,
      "status": "available",
      "volumeId": 12
    },
    {
      "createdAt": "2013-02-25T16:27:54.685008",
      "displayDescription": "Default description",
      "displayName": "Default name",
      "id": 102,
      "size": 100,
      "status": "available",
      "volumeId": 12
    }
  ]
}
```

4.52.10. List snapshots with details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-snapshots/detail	Lists all snapshots with details.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.10.1. Request

This table shows the URI parameters for the list snapshots with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

4.52.10.2. Response

Example 4.269. List snapshots with details: JSON response

```
{
  "snapshots": [
    {
      "createdAt": "2013-02-25T16:27:54.684999",
      "displayDescription": "Default description",
      "displayName": "Default name",
      "id": 100,
      "size": 100,
      "status": "available",
      "volumeId": 12
    },
    {
      "createdAt": "2013-02-25T16:27:54.685005",
      "displayDescription": "Default description",
      "displayName": "Default name",
      "id": 101,
      "size": 100,
      "status": "available",
      "volumeId": 12
    },
    {
      "createdAt": "2013-02-25T16:27:54.685008",
      "displayDescription": "Default description",
      "displayName": "Default name",
      "id": 102,
      "size": 100,
      "status": "available",
      "volumeId": 12
    }
  ]
}
```


4.52.11. Show snapshot details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-snapshots/{snapshot_id}	Shows details for a snapshot.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.11.1. Request

This table shows the URI parameters for the show snapshot details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

4.52.11.2. Response

Example 4.270. Show snapshot details: JSON response

```
{
  "snapshot": {
    "createdAt": "2013-02-25T16:27:54.724209",
    "displayDescription": "Default description",
    "displayName": "Default name",
    "id": "100",
    "size": 100,
    "status": "available",
    "volumeId": 12
  }
}
```

4.52.12. Delete snapshot

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-snapshots/{snapshot_id}	Deletes a snapshot from the account.

This operation is asynchronous. You must list snapshots repeatedly to determine whether the snapshot was deleted.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404)

4.52.12.1. Request

This table shows the URI parameters for the delete snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

4.52.12.2. Response

Example 4.271. Delete snapshot: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Date: Mon, 01 Dec 2014 16:23:10 GMT
```

This operation does not return a response body.

5. Database Service API v1.0 (CURRENT)

Method	URI	Description
API versions		
GET	/	Lists information about all Database Service API versions.
GET	/v1.0	Shows details for the Database Service API v1.0.
Database instances (instances)		
POST	/v1.0/{accountId}/instances	Creates a database instance.
GET	/v1.0/{accountId}/instances	Lists information, including status, for all database instances.
GET	/v1.0/{accountId}/instances/{instanceId}	Shows database instance details.
PUT	/v1.0/{accountId}/instances/{instanceId}	Attaches a configuration group to an instance.
PUT	/v1.0/{accountId}/instances/{instanceId}	Detaches a configuration group from an instance.
DELETE	/v1.0/{accountId}/instances/{instanceId}	Deletes a database instance, including any associated data.
PATCH	/v1.0/{accountId}/instances/{instanceId}	Detaches a replica from its replication source.
GET	/v1.0/{accountId}/instances/{instanceId}/configuration	Lists the configuration defaults for an instance.
Database instance actions (action)		
POST	/v1.0/{accountId}/instances/{instanceId}/action	Resizes the memory for an instance.
POST	/v1.0/{accountId}/instances/{instanceId}/action	Resizes the volume that is attached to an instance.
POST	/v1.0/{accountId}/instances/{instanceId}/action	Restarts the database service for an instance.
POST	/v1.0/{accountId}/instances/{instanceId}/action	Promotes a replica.
POST	/v1.0/{accountId}/instances/{instanceId}/action	Deletes the base instance in a replication set.
Databases (databases)		
POST	/v1.0/{accountId}/instances/{instanceId}/databases	Creates a database within an instance.
GET	/v1.0/{accountId}/instances/{instanceId}/databases	Lists databases for an instance.
DELETE	/v1.0/{accountId}/instances/{instanceId}/databases/{databaseName}	Deletes a database.
Users (users)		
POST	/v1.0/{accountId}/instances/{instanceId}/users	Creates a user for a database instance.
GET	/v1.0/{accountId}/instances/{instanceId}/users	Lists the users in a database instance and the associated databases for that user.
DELETE	/v1.0/{accountId}/instances/{instanceId}/users/{name}	Deletes a user for a database instance.
POST	/v1.0/{accountId}/instances/{instanceId}/root	Enables the root user for a database instance and returns the root password.
GET	/v1.0/{accountId}/instances/{instanceId}/root	Shows root-enabled status for a database instance.

Method	URI	Description
DELETE	/v1.0/{accountId}/instances/{instanceId}/root	Disables the root user.
Flavors (flavors)		
GET	/v1.0/{accountId}/flavors	Lists information for all available flavors.
GET	/v1.0/{accountId}/flavors/{flavorId}	Shows flavor details with details of the RAM.
Data stores (datastores)		
GET	/v1.0/{accountId}/datastores/{datastore_name}/versions	Lists the available versions of a data store.
GET	/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters	Lists the available configuration parameters for a data store version.
GET	/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters/{parameter_name}	Displays details for a configuration parameter associated with a data store version.
Configuration groups (configurations)		
POST	/v1.0/{accountId}/configurations	Creates a configuration group.
GET	/v1.0/{accountId}/configurations	Lists all configuration groups.
DELETE	/v1.0/{accountId}/configurations/{configId}	Deletes a configuration group.
PATCH	/v1.0/{accountId}/configurations/{configId}	Sets new values for a configuration group.
GET	/v1.0/{accountId}/configurations/{configId}	Lists details about a configuration group, including its values.
PUT	/v1.0/{accountId}/configurations/{configId}	Sets new values for a configuration group. Also lets you change the name and description of the configuration group.
GET	/v1.0/{accountId}/configurations/{configId}/instances	Lists the instances associated with the specified configuration group.

5.1. API versions

Lists information for all Database Service API versions and shows Database Service v1.0 details.

Method	URI	Description
GET	/	Lists information about all Database Service API versions.
GET	/v1.0	Shows details for the Database Service API v1.0.

5.1.1. List versions

Method	URI	Description
GET	/	Lists information about all Database Service API versions.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.1.1.1. Request

Example 5.1. List versions: JSON request

```
GET / HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.1.1.2. Response

Example 5.2. List versions: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 153
Date: Wed, 25 Jan 2012 21:53:04 GMT
```

```
{
  "versions": [
    {
      "id": "v1.0",
      "links": [
        {
          "href": "https://openstack.example.com/v1.0/",
          "rel": "self"
        }
      ],
      "status": "CURRENT",
      "updated": "2012-01-01T00:00:00Z"
    }
  ]
}
```

5.1.2. Show version details

Method	URI	Description
GET	/v1.0	Shows details for the Database Service API v1.0.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.1.2.1. Request

Example 5.3. Show version details: JSON request

```
GET /v1.0/ HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.1.2.2. Response

Example 5.4. Show version details: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 158
Date: Tue, 22 Apr 2014 19:02:58 GMT
```

```
{
  "versions": [
    {
      "status": "CURRENT",
      "updated": "2012-08-01T00:00:00Z",
      "id": "v1.0",
      "links": [
        {
          "href": "http://23.253.228.211:8779/v1.0/",
          "rel": "self"
        }
      ]
    }
  ]
}
```

5.2. Database instances (instances)

Creates, lists, shows details for, attaches a configuration group to, detaches a configuration group from, deletes, lists configuration defaults, creates root, and determines whether root is enables for instances.

Method	URI	Description
POST	/v1.0/{accountId}/instances	Creates a database instance.
GET	/v1.0/{accountId}/instances	Lists information, including status, for all database instances.
GET	/v1.0/{accountId}/instances/{instanceId}	Shows database instance details.
PUT	/v1.0/{accountId}/instances/{instanceId}	Attaches a configuration group to an instance.
PUT	/v1.0/{accountId}/instances/{instanceId}	Detaches a configuration group from an instance.
DELETE	/v1.0/{accountId}/instances/{instanceId}	Deletes a database instance, including any associated data.
PATCH	/v1.0/{accountId}/instances/{instanceId}	Detaches a replica from its replication source.
GET	/v1.0/{accountId}/instances/{instanceId}/configuration	Lists the configuration defaults for an instance.

5.2.1. Create database instance

Method	URI	Description
POST	/v1.0/{accountId}/instances	Creates a database instance.

Asynchronously provisions a database instance. You must specify a flavor and a volume size. The service provisions the instance with a volume of the requested size, which serves as storage for the database instance.



Notes

- You can create only one database instance per **POST** request.
- You can create a database instance with one or more databases. You associate users with each database.
- The default binding for the MySQL instance is port 3306.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.2.1.1. Request

This table shows the URI parameters for the create database instance request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.

Example 5.5. Create database instance: JSON request

```
POST /v1.0/1234/instances HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: d6cafa5b-e0c7-4ab8-948e-7c95f2acd031
Accept: application/json
Content-Type: application/json
```

```
{
  "instance": {
    "databases": [
      {
        "character_set": "utf8",
        "collate": "utf8_general_ci",
        "name": "sampledb"
      },
      {
        "name": "nextround"
      }
    ],
    "flavorRef": "https://openstack.example.com/v1.0/1234/flavors/1",
    "name": "json Rack instance",
```

```

    "users": [
      {
        "databases": [
          {
            "name": "sampledb"
          }
        ],
        "name": "demouser",
        "password": "secretsecret"
      }
    ],
    "volume": {
      "size": 2
    }
  }
}

```

5.2.1.2. Response

Example 5.6. Create database instance: JSON response

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 636
Date: Wed, 25 Jan 2012 21:53:10 GMT

```

```

{
  "instance": {
    "created": "2012-01-25T21:53:09Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "https://openstack.example.com/v1.0/1234/flavors/1",
          "rel": "self"
        },
        {
          "href": "https://openstack.example.com/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostname": "e09ad9a3f73309469cf1f43d11e79549caf9acf2.rackspacecloud.db.com",
    "id": "dea5a2f7-3ec7-4496-adab-0abb5a42d635",
    "links": [
      {
        "href": "https://openstack.example.com/v1.0/1234/instances/dea5a2f7-3ec7-4496-adab-0abb5a42d635",
        "rel": "self"
      },
      {
        "href": "https://openstack.example.com/instances/dea5a2f7-3ec7-4496-adab-0abb5a42d635",
        "rel": "bookmark"
      }
    ],
    "name": "json_rack_instance",
    "status": "BUILD",
  }
}

```

```
    "updated": "2012-01-25T21:53:10Z",  
    "volume": {  
      "size": 2  
    }  
  }  
}
```

The previous response examples show resources that contain links to themselves that enable a client to easily obtain resource URIs rather than construct them. Resources can have these link relations:

- A `self` link contains a *versioned* link to the resource. Use these links in cases where the link will be followed immediately.
- A `bookmark` link provides a permanent link to a resource that is appropriate for long-term storage.

5.2.2. List database instances

Method	URI	Description
GET	/v1.0/{accountId}/instances	Lists information, including status, for all database instances.

Lists status and information for all database instances.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.2.2.1. Request

This table shows the URI parameters for the list database instances request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.

Example 5.7. List database instances: JSON request

```
GET /v1.0/1234/instances HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.2.2.2. Response

Example 5.8. List database instances: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1150
Date: Tue, 19 Jun 2012 19:53:04 GMT
```

```
{
  "instances": [
    {
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "https://openstack.example.com/v1.0/1234/flavors/1",
            "rel": "self"
          },
          {
            "href": "https://openstack.example.com/flavors/1",
```

```

        "rel": "bookmark"
    }
    ],
    },
    "id": "28d1b8f3-172a-4f6d-983d-36021508444a",
    "links": [
        {
            "href": "https://openstack.example.com/v1.0/1234/instances/28d1b8f3-172a-4f6d-983d-36021508444a",
            "rel": "self"
        },
        {
            "href": "https://openstack.example.com/instances/28d1b8f3-172a-4f6d-983d-36021508444a",
            "rel": "bookmark"
        }
    ],
    "name": "json_rack_instance",
    "status": "ACTIVE",
    "volume": {
        "size": 2
    }
},
{
    "flavor": {
        "id": "1",
        "links": [
            {
                "href": "https://openstack.example.com/v1.0/1234/flavors/1",
                "rel": "self"
            },
            {
                "href": "https://openstack.example.com/flavors/1",
                "rel": "bookmark"
            }
        ]
    },
    "id": "8fb081af-f237-44f5-80cc-b46be1840ca9",
    "links": [
        {
            "href": "https://openstack.example.com/v1.0/1234/instances/8fb081af-f237-44f5-80cc-b46be1840ca9",
            "rel": "self"
        },
        {
            "href": "https://openstack.example.com/instances/8fb081af-f237-44f5-80cc-b46be1840ca9",
            "rel": "bookmark"
        }
    ],
    "name": "xml_rack_instance",
    "status": "ACTIVE",
    "volume": {
        "size": 2
    }
}
]
}

```

5.2.3. Show database instance details

Method	URI	Description
GET	/v1.0/{accountId}/instances/{instanceId}	Shows database instance details.

Lists the status and details of the database instance.

Lists the volume size in gigabytes (GB) and the approximate GB used.



Note

After instance creation, the used value is greater than 0, which is expected and due to the automatic creation of non-empty transaction logs for MySQL optimization. The response does not include the used attribute when the instance status is BUILD, REBOOT, RESIZE, or ERROR.

The list operations return a DNS-resolvable host name for the database instance rather than an IP address. Because the host name always resolves to the correct IP address for the database instance, you do not need to maintain the mapping. Although the IP address might change when you resize, migrate, or perform other operations, the host name always resolves to the correct database instance.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.2.3.1. Request

This table shows the URI parameters for the show database instance details request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.9. Show database instance details: JSON request

```
GET /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.2.3.2. Response

Example 5.10. Show database instance details: JSON response

```
HTTP/1.1 200 OK
```

Content-Type: application/json
Content-Length: 685
Date: Wed, 28 Mar 2012 21:37:29 GMT

```
{
  "instance": {
    "created": "2012-03-28T21:31:02Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "https://openstack.example.com/v1.0/1234/flavors/1",
          "rel": "self"
        },
        {
          "href": "https://openstack.example.com/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostname": "e09ad9a3f73309469cf1f43d11e79549caf9acf2.rackspacecloud.db.com",
    "id": "2450c73f-7805-4afe-a42c-4094ab42666b",
    "links": [
      {
        "href": "https://openstack.example.com/v1.0/1234/instances/2450c73f-7805-4afe-a42c-4094ab42666b",
        "rel": "self"
      },
      {
        "href": "https://openstack.example.com/instances/2450c73f-7805-4afe-a42c-4094ab42666b",
        "rel": "bookmark"
      }
    ],
    "name": "xml_rack_instance",
    "status": "ACTIVE",
    "updated": "2012-03-28T21:34:25Z",
    "volume": {
      "size": 2,
      "used": 0.124542236328125
    }
  }
}
```

5.2.4. Attach configuration group

Method	URI	Description
PUT	/v1.0/{accountId}/instances/{instanceId}	Attaches a configuration group to an instance.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

5.2.4.1. Request

This table shows the URI parameters for the attach configuration group request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.11. Attach configuration group: JSON request

```
PUT /v1.0/1234/instances/4c93c73b-d6d0-47d7-b8c6-b699d19d7de9 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
  "instance": {
    "configuration": "2aa51628-5c42-4086-8682-137caffd2ba6"
  }
}
```

5.2.4.2. Response

Example 5.12. Attach configuration group: JSON response

```
HTTP/1.1 202 OK
Content-Type: application/json
Content-Length: 0
Date: Mon, 13 Jul 2015 19:53:04 GMT
```

This operation does not return a response body.

5.2.5. Detach configuration group

Method	URI	Description
PUT	/v1.0/{accountId}/instances/{instanceId}	Detaches a configuration group from an instance.

When you pass in only an instance ID and omit the configuration ID, this operation detaches any configuration group that was attached to the instance.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

5.2.5.1. Request

This table shows the URI parameters for the detach configuration group request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.13. Detach configuration group: JSON request

```
PUT /v1.0/1234/instances/4c93c73b-d6d0-47d7-b8c6-b699d19d7de9 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
  "instance": {}
}
```

5.2.5.2. Response

Example 5.14. Detach configuration group: JSON response

```
HTTP/1.1 202 OK
Content-Type: application/json
Content-Length: 0
Date: Mon, 13 Jul 2015 19:53:04 GMT
```

This operation does not return a response body.

5.2.6. Delete database instance

Method	URI	Description
DELETE	/v1.0/{accountId}/instances/{instanceId}	Deletes a database instance, including any associated data.



Note

This operation does not delete any read slaves.



Note

You cannot complete this operation when the instance state is either RE-BUILDING or BUILDING.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), 422, itemNotFound (404)

5.2.6.1. Request

This table shows the URI parameters for the delete database instance request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.15. Delete database instance: JSON request

```
DELETE /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.2.7. Detach replica

Method	URI	Description
PATCH	/v1.0/{accountId}/instances/{instanceId}	Detaches a replica from its replication source.

If you created an instance that is a replica of a source instance, you can detach the replica from the source. This can be useful if the source becomes unavailable. In this case, you can detach the replica from the source, making the replica a standalone database instance. You can then take the new standalone instance and create a new replica of that instance.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

5.2.7.1. Request

This table shows the URI parameters for the detach replica request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.16. Detach replica: JSON request

```
PATCH /v1.0/1234/instances/4c93c73b-d6d0-47d7-b8c6-b699d19d7de9 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
  "instance": {
    "replica_of": null,
    "slave_of": null
  }
}
```

5.2.7.2. Response

Example 5.17. Detach replica: JSON response

```
HTTP/1.1 202 OK
Content-Type: application/json
Content-Length: 0
Date: Mon, 13 Jul 2015 19:53:04 GMT
```

This operation does not return a response body.

5.2.8. List configuration defaults

Method	URI	Description
GET	/v1.0/{accountId}/instances/{instanceId}/configuration	Lists the configuration defaults for an instance.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.2.8.1. Request

This table shows the URI parameters for the list configuration defaults request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.18. List configuration defaults: JSON request

```
GET /v1.0/1234/instances/instance_1 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.2.8.2. Response

Example 5.19. List configuration defaults: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1559
Date: Wed, 08 Jul 2015 19:53:04 GMT
```

```
{
  "instance": {
    "configuration": {
      "tmp_table_size": "16M",
      "innodb_log_files_in_group": "2",
      "skip-external-locking": "1",
      "read_rnd_buffer_size": "512 K",
      "max_user_connections": "100",
      "max_heap_table_size": "16M",
      "port": "3306",
      "tmpdir": "/var/tmp",
      "pid_file": "/var/run/mysqld/mysqld.pid",
```

```

    "myisam-recover ": "BACKUP",
    "server_id": "334596",
    "innodb_buffer_pool_size": "150M",
    "basedir": "/usr",
    "max_allowed_packet": "1024K",
    "datadir": "/var/lib/mysql/data",
    "innodb_log_buffer_size": "25M",
    "max_connections": "100",
    "table_open_cache": "256",
    "connect_timeout": "15",
    "query_cache_type": "1",
    "local-infile": "0",
    "innodb_log_file_size": "50M",
    "thread_stack": "192K",
    "query_cache_limit": "1M",
    "wait_timeout": "120",
    "user": "mysql",
    "thread_cache_size": "4",
    "query_cache_size": "8M",
    "innodb_data_file_path": "ibdata1:10M:autoextend",
    "default_storage_engine": "innodb",
    "sort_buffer_size": "1M",
    "table_definition_cache": "256",
    "read_buffer_size": "512K",
    "open_files_limit": "512",
    "innodb_file_per_table": "1",
    "key_buffer_size": "50M",
    "join_buffer_size": "1M"
  }
}

```

5.3. Database instance actions (action)

Resizes instances and volumes and restarts instances.

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/action	Resizes the memory for an instance.
POST	/v1.0/{accountId}/instances/{instanceId}/action	Resizes the volume that is attached to an instance.
POST	/v1.0/{accountId}/instances/{instanceId}/action	Restarts the database service for an instance.
POST	/v1.0/{accountId}/instances/{instanceId}/action	Promotes a replica.
POST	/v1.0/{accountId}/instances/{instanceId}/action	Deletes the base instance in a replication set.

5.3.1. Resize instance

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/action	Resizes the memory for an instance.

If you provide a valid `flavorRef`, this operation changes the memory size of the instance, and restarts MySQL.

Normal response codes: 202

Error response codes: `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `overLimit` (413), `unprocessableEntity` (422), `instanceFault` (500), `notImplemented` (501), `serviceUnavailable` (503), `itemNotFound` (404), `badMediaType` (415)

5.3.1.1. Request

This table shows the URI parameters for the resize instance request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.20. Resize instance: JSON request

```
POST /v1.0/1234/instances/23a3d4fb-3731-497b-afd4-bf25bde2b5fc/action HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 2eeb3252-0164-40f5-8fb7-85df5faa2698
Accept: application/json
Content-Type: application/json
```

```
{
  "resize": {
    "flavorRef": "https://openstack.example.com/v1.0/1234/flavors/2"
  }
}
```

5.3.1.2. Response

Example 5.21. Resize instance: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: text/plain; charset=UTF-8
Content-Length: 58
Date: Mon, 06 Feb 2012 21:28:10 GMT
```

This operation does not return a response body.

5.3.2. Resize instance volume

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/action	Resizes the volume that is attached to an instance.

You can use this operation to increase but not decrease the volume size. A valid volume size is an integer value in gigabytes (GB).



Note

You cannot increase the volume to a size that is larger than the API volume size limit.

If this operation succeeds, it returns a 202 Accepted response.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

5.3.2.1. Request

This table shows the URI parameters for the resize instance volume request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.22. Resize instance volume: JSON request

```
POST /v1.0/1234/instances/23a3d4fb-3731-497b-afd4-bf25bde2b5fc/action HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 2eeb3252-0164-40f5-8fb7-85df5faa2698
Accept: application/json
Content-Type: application/json
```

```
{
  "resize": {
    "volume": {
      "size": 4
    }
  }
}
```

5.3.2.2. Response

Example 5.23. Resize instance volume: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
```

```
Content-Length: 0  
Date: Wed, 27 Jun 2012 23:12:20 GMT
```

This operation does not return a response body.

5.3.3. Restart instance

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/action	Restarts the database service for an instance.

The restart operation restarts only the MySQL instance. Restarting MySQL erases any dynamic configuration settings that you make in MySQL.



Note

The MySQL service is unavailable until the instance restarts.

If the operation succeeds, it returns the Accepted (202) response code.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

5.3.3.1. Request

This table shows the URI parameters for the restart instance request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.24. Restart instance: JSON request

```
POST /v1.0/1234/instances/13d940c4-70bb-4ff4-8866-6ee9ab5e5cae/action HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
  "restart": {}
}
```

5.3.3.2. Response

Example 5.25. Restart instance: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:19 GMT
```

This operation does not return a response body.

5.3.4. Promote instance to replica source

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/action	Promotes a replica.

If you have set up replication, and the base instance is still reachable, you can use this operation to promote a replica to be the new base instance.

This can be useful if you want to make a configuration change to the base instance that your replicas are replicating from. For example, you might want to increase the disk or CPU capacity. If you made the change on the base instance directly, you would need to take the base instance down for the duration of the operation. Instead, you can create a replica, make the configuration change on the replica, and then promote the replica to become the new base instance.

For `instanceId`, pass in the instance ID of the replica you want to promote.

Normal response codes: 202

Error response codes: `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `overLimit` (413), `unprocessableEntity` (422), `instanceFault` (500), `notImplemented` (501), `serviceUnavailable` (503), `itemNotFound` (404), `badMediaType` (415)

5.3.4.1. Request

This table shows the URI parameters for the promote instance to replica source request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.26. Promote instance to replica source: JSON request

```
POST /v1.0/1234/instances/13d940c4-70bb-4ff4-8866-6ee9ab5e5cae/action HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
  "promote_to_replica_source": {}
}
```

5.3.4.2. Response

Example 5.27. Promote instance to replica source: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Fri, 08 Apr 2016 23:11:19 GMT
```

This operation does not return a response body.

5.3.5. Delete replication base instance

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/action	Deletes the base instance in a replication set.

If the base instance becomes unreachable, you can use this operation to delete the base instance.

This operation:

- Finds the replica that has processed the greatest number of transactions and picks that replica to use as the new base instance.
- Transfers the public IP of the old base instance to the new base instance (which is the newly-promoted replica).
- Deletes the old base instance.
- Takes all the instances in the replication set and makes them start replicating from the new base instance.

For `instanceId`, pass in the instance ID of the unreachable base instance.

Normal response codes: 202

Error response codes: `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `overLimit` (413), `unprocessableEntity` (422), `instanceFault` (500), `notImplemented` (501), `serviceUnavailable` (503), `itemNotFound` (404), `badMediaType` (415)

5.3.5.1. Request

This table shows the URI parameters for the delete replication base instance request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.28. Delete replication base instance: JSON request

```
POST /v1.0/1234/instances/13d940c4-70bb-4ff4-8866-6ee9ab5e5cae/action HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
  "eject_replica_source": {}
}
```

5.3.5.2. Response

Example 5.29. Delete replication base instance: JSON response

```
HTTP/1.1 202 Accepted
```

```
Content-Type: application/json
Content-Length: 0
Date: Fri, 08 Apr 2016 23:11:19 GMT
```

This operation does not return a response body.

5.4. Databases (databases)

Creates, lists all, and deletes databases.

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/databases	Creates a database within an instance.
GET	/v1.0/{accountId}/instances/{instanceId}/databases	Lists databases for an instance.
DELETE	/v1.0/{accountId}/instances/{instanceId}/databases/{databaseName}	Deletes a database.

5.4.1. Create database

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/databases	Creates a database within an instance.

The name of the database is a required attribute.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.4.1.1. Request

This table shows the URI parameters for the create database request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.30. Create database: JSON request

```
POST /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/databases HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
  "databases": [
    {
      "character_set": "utf8",
      "collate": "utf8_general_ci",
      "name": "testingdb"
    },
    {
      "name": "sampledb"
    }
  ]
}
```

5.4.1.2. Response

Example 5.31. Create database: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:18 GMT
```

This operation does not return a response body.

5.4.2. List instance databases

Method	URI	Description
GET	/v1.0/{accountId}/instances/{instanceId}/databases	Lists databases for an instance.



Note

This operation returns only the user-defined databases and not the system databases. Only the database administrator can view the `mysql`, `information_schema`, and `lost+found` system databases.

Normal response codes: 200

Error response codes: `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `overLimit` (413), `unprocessableEntity` (422), `instanceFault` (500), `notImplemented` (501), `serviceUnavailable` (503), `itemNotFound` (404)

5.4.2.1. Request

This table shows the URI parameters for the list instance databases request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.32. List instance databases: JSON request

```
GET /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/databases HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.4.2.2. Response

Example 5.33. List instance databases: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 136
Date: Wed, 25 Jan 2012 21:58:01 GMT
```

```
{
  "databases": [
    {
      "name": "anotherexampleldb"
    },
    {
```

```
        "name": "exampledb"
      },
      {
        "name": "nextround"
      },
      {
        "name": "sampledb"
      },
      {
        "name": "testingdb"
      }
    ]
  }
```


5.4.3. Delete database

Method	URI	Description
DELETE	/v1.0/{accountId}/instances/{instanceId}/databases/{databaseName}	Deletes a database.

This operation also deletes all data that is associated with the database.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.4.3.1. Request

This table shows the URI parameters for the delete database request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.
{databaseName}	String	The name for the database.

Example 5.34. Delete database: JSON request

```
DELETE /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/databases/
exampledb HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.4.3.2. Response

Example 5.35. Delete database: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:18 GMT
```

This operation does not return a response body.

5.5. Users (users)

Creates, lists all, and deletes users.

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/users	Creates a user for a database instance.
GET	/v1.0/{accountId}/instances/{instanceId}/users	Lists the users in a database instance and the associated databases for that user.
DELETE	/v1.0/{accountId}/instances/{instanceId}/users/{name}	Deletes a user for a database instance.
POST	/v1.0/{accountId}/instances/{instanceId}/root	Enables the root user for a database instance and returns the root password.
GET	/v1.0/{accountId}/instances/{instanceId}/root	Shows root-enabled status for a database instance.
DELETE	/v1.0/{accountId}/instances/{instanceId}/root	Disables the root user.

5.5.1. Create user

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/users	Creates a user for a database instance.

Asynchronously provisions a new user for the database instance by using the configuration that you define in the request object. After the API validates the request and starts progress on the provisioning process, the call returns the Accepted (202) response code.

If the API cannot fulfill the corresponding request due to insufficient data or data that is not valid, the API returns the Bad Request (400) response code with information about the nature of the failure. You cannot recover from validation errors. You must correct the cause of the failure and the request again.

This table lists the required attributes for creating users:

Table 5.1. Required attributes for user

Applies to	Name	Description	Required
User	name	Name of the user for the database.	Yes
	password	User password for database access.	Yes
	(database) name	Name of the database that the user can access. You must specify one or more database names.	No



Notes

- The operation grants the user all privileges on the databases.
- Do not use the root user name, which is reserved.

These tables list the valid characters for database names, user names, and passwords.

Table 5.2. Valid characters in database name, user name, and password

Character
Letters (upper and lower cases allowed)
Numbers
@, ?, #, and spaces are allowed, but <i>not</i> at the beginning and end of the database name, user name, and password
_ is allowed anywhere in the database name, user name, and password

Table 5.3. Characters that are not allowed in database name, user name, and password

Character
Single quotes

Character
Double quotes
Back quotes
Semicolons
Commas
Back slashes
Forward slashes
Spaces at the front or end of the user name or password

Table 5.4. Length restrictions for database name, user name, and password

Restriction	Value
Database name maximum length	64
User name maximum length	16
Password maximum length	unlimited (no restrictions)

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.5.1.1. Request

This table shows the URI parameters for the create user request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.36. Create user: JSON request

```
POST /v1.0/1234/instances/1c59bdb8-03b6-4079-a7db-ba92d23a98b3/users HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: bb64d788-2dec-4a6b-a670-7151d108cacf
Accept: application/json
Content-Type: application/json
```

```
{
  "users": [
    {
      "databases": [
        {
          "name": "databaseA"
        }
      ],
      "name": "dbuser3",
      "password": "secretsecret"
    },
    {
      "databases": [
        {
```

```
        "name": "databaseB"
      },
      {
        "name": "databaseC"
      }
    ],
    "name": "dbuser4",
    "password": "secretsecret"
  }
}
```

5.5.1.2. Response

Example 5.37. Create user: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:18 GMT
```

This operation does not return a response body.

5.5.2. List database instance users

Method	URI	Description
GET	/v1.0/{accountId}/instances/{instanceId}/users	Lists the users in a database instance and the associated databases for that user.



Note

This operation does not return system users. A system user is a database administrator who administers the health of the database. Also, this operation returns the root user only if it is enabled.

The following notes apply to MySQL users:

- User names can be up to 16 characters long.
- When you create accounts with INSERT, you must use FLUSH PRIVILEGES to tell the server to reload the grant tables.
- For additional information, See: <http://dev.mysql.com/doc/refman/5.1/en/user-account-management.html>

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.5.2.1. Request

This table shows the URI parameters for the list database instance users request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.38. List database instance users: JSON request

```
GET /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/users HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.5.2.2. Response

Example 5.39. List database instance users: JSON response

```
HTTP/1.1 200 OK
```

Content-Type: application/json
Content-Length: 152
Date: Wed, 21 Mar 2012 17:46:46 GMT

```
{
  "users": [
    {
      "databases": [
        {
          "name": "databaseA"
        }
      ],
      "name": "dbuser3"
    },
    {
      "databases": [
        {
          "name": "databaseB"
        },
        {
          "name": "databaseC"
        }
      ],
      "name": "dbuser4"
    }
  ]
}
```

5.5.3. Delete user

Method	URI	Description
DELETE	/v1.0/{accountId}/instances/{instanceId}/users/{name}	Deletes a user for a database instance.



Warning

Do not use periods in user names. A bug in a Python library that Rackspace uses that can cause incorrect user deletions to occur if you use a period (.) in the user name. In this case, the bug in the library truncates the user name to the portion from the beginning up to the period. For example, for the my.userA user, the bug truncates the user name to my, and if the user exists, that user is incorrectly deleted.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.5.3.1. Request

This table shows the URI parameters for the delete user request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.
{userName}	String	The name for the user.

Example 5.40. Delete user: JSON request

```
DELETE /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/users/
testuser HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.5.3.2. Response

Example 5.41. Delete user: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:19 GMT
```


This operation does not return a response body.

5.5.4. Enable root user

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/root	Enables the root user for a database instance and returns the root password.

This operation generates a root password for the root user and enables the root user to log in from any host.



Note

Changes that you make as a root user can impact the database instance and API operations in unpredictable and detrimental ways. When you enable the root user, you accept the possibility that we cannot support your database instance. We might not be able to assist you if you change core MySQL settings. These changes can be, but are not limited to, turning off bin logs, removing users that we use to access your instance, and so on.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.5.4.1. Request

This table shows the URI parameters for the enable root user request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.42. Enable root user: JSON request

```
POST /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/root HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.5.4.2. Response

Example 5.43. Enable root user: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 78
```

Date: Wed, 25 Jan 2012 21:58:11 GMT

```
{
  "user": {
    "name": "root",
    "password": "secretsecret"
  }
}
```

5.5.5. Show root-enabled status for database instance

Method	URI	Description
GET	/v1.0/{accountId}/instances/{instanceId}/root	Shows root-enabled status for a database instance.

Returns `true` if root user is enabled for a database instance. Otherwise, returns `false`.

Normal response codes: 200

Error response codes: `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `overLimit` (413), `unprocessableEntity` (422), `instanceFault` (500), `notImplemented` (501), `serviceUnavailable` (503), `itemNotFound` (404)

5.5.5.1. Request

This table shows the URI parameters for the show root-enabled status for database instance request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.44. Show root-enabled status for database instance: JSON request

```
GET /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/root HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.5.5.2. Response

Example 5.45. Show root-enabled status for database instance: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 21
Date: Wed, 25 Jan 2012 21:58:13 GMT
```

```
{
  "rootEnabled": true
}
```

5.5.6. Disable root user

Method	URI	Description
DELETE	/v1.0/{accountId}/instances/{instanceId}/root	Disables the root user.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.5.6.1. Request

This table shows the URI parameters for the disable root user request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{instanceId}	String	The ID for the database instance.

Example 5.46. Disable root user: JSON request

```
DELETE /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/root HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.5.6.2. Response

Example 5.47. Disable root user: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:18 GMT
```

This operation does not return a response body.

5.6. Flavors (flavors)

Lists all flavors and shows details for a flavor, by ID.

Method	URI	Description
GET	/v1.0/{accountId}/flavors	Lists information for all available flavors.
GET	/v1.0/{accountId}/flavors/{flavorId}	Shows flavor details with details of the RAM.

5.6.1. List flavors

Method	URI	Description
GET	/v1.0/{accountId}/flavors	Lists information for all available flavors.

This operation lists information for all available flavors.

This resource is identical to the flavors found in the OpenStack Nova API, but without the disk property.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.6.1.1. Request

This table shows the URI parameters for the list flavors request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.

Example 5.48. List flavors: JSON request

```
GET /v1.0/1234/flavors HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.6.1.2. Response

Example 5.49. List flavors: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1768
Date: Tue, 19 Jun 2012 19:52:45 GMT
```

```
{
  "flavors": [
    {
      "id": 1,
      "links": [
        {
          "href": "https://openstack.example.com/v1.0/1234/flavors/1",
          "rel": "self"
        }
      ]
    }
  ]
}
```

```

        },
        {
            "href": "https://openstack.example.com/flavors/1",
            "rel": "bookmark"
        }
    ],
    "name": "m1.tiny",
    "ram": 512
},
{
    "id": 2,
    "links": [
        {
            "href": "https://openstack.example.com/v1.0/1234/flavors/
2",
            "rel": "self"
        },
        {
            "href": "https://openstack.example.com/flavors/2",
            "rel": "bookmark"
        }
    ],
    "name": "m1.small",
    "ram": 1024
},
{
    "id": 3,
    "links": [
        {
            "href": "https://openstack.example.com/v1.0/1234/flavors/
3",
            "rel": "self"
        },
        {
            "href": "https://openstack.example.com/flavors/3",
            "rel": "bookmark"
        }
    ],
    "name": "m1.medium",
    "ram": 2048
},
{
    "id": 4,
    "links": [
        {
            "href": "https://openstack.example.com/v1.0/1234/flavors/
4",
            "rel": "self"
        },
        {
            "href": "https://openstack.example.com/flavors/4",
            "rel": "bookmark"
        }
    ],
    "name": "m1.large",
    "ram": 4096
    }
    ]
}

```


5.6.2. Show flavor details

Method	URI	Description
GET	/v1.0/{accountId}/flavors/{flavorId}	Shows flavor details with details of the RAM.

This resource is identical to the flavors found in the OpenStack Compute API, but without the disk property.



Note

The `flavorId` parameter must be an integer value. If you use a floating point value for this parameter, this call truncates the decimal portion and uses the integer portion as the `flavorId` value.

Normal response codes: 200

Error response codes: `badRequest` (400), `unauthorized` (401), `forbidden` (403), `badMethod` (405), `overLimit` (413), `unprocessableEntity` (422), `instanceFault` (500), `notImplemented` (501), `serviceUnavailable` (503), `itemNotFound` (404)

5.6.2.1. Request

This table shows the URI parameters for the show flavor details request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{flavorId}	String	The ID for the flavor.

Example 5.50. Show flavor details: JSON request

```
GET /v1.0/1234/flavors/1 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.6.2.2. Response

Example 5.51. Show flavor details: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 209
Date: Wed, 25 Jan 2012 21:53:05 GMT
```

```
{
  "flavor": {
```

```
{
  "id": 1,
  "links": [
    {
      "href": "https://openstack.example.com/v1.0/1234/flavors/1",
      "rel": "self"
    },
    {
      "href": "https://openstack.example.com/flavors/1",
      "rel": "bookmark"
    }
  ],
  "name": "m1.tiny",
  "ram": 512
}
```

5.7. Data stores (datastores)

Lists data store versions, lists parameters for data stores, and shows parameter details for a data store version.

Method	URI	Description
GET	/v1.0/{accountId}/datastores/{datastore_name}/versions	Lists the available versions of a data store.
GET	/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters	Lists the available configuration parameters for a data store version.
GET	/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters/{parameter_name}	Displays details for a configuration parameter associated with a data store version.

5.7.1. List datastore versions

Method	URI	Description
GET	/v1.0/{accountId}/datastores/{datastore_name}/versions	Lists the available versions of a data store.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.7.1.1. Request

This table shows the URI parameters for the list datastore versions request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{datastore_name}	String	The name of the data store.

Example 5.52. List datastore versions: JSON request

```
GET /v1.0/1234/datastores/mysql/versions HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.7.1.2. Response

Example 5.53. List datastore versions: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 580
Date: Tue, 23 Jun 2015 21:58:13 GMT
```

```
{
  "name": "5.6",
  "links": [
    {
      "href": "https://10.240.28.38:8779/v1.0/27bee406abb5486e81ef3ff4382aabaf/datastores/versions/2dc7faa0-efff-4c2b-8cff-bcd949c518a5",
      "rel": "self"
    },
    {
      "href": "https://10.240.28.38:8779/datastores/versions/2dc7faa0-efff-4c2b-8cff-bcd949c518a5",
      "rel": "bookmark"
    }
  ]
}
```

```
],  
  "image": "b69fbd9e-b31d-46ff-8afb-cbf452f6f835",  
  "active": 1,  
  "datastore": "3a8968d8-e5f5-4452-83ca-f6c90b5de06a",  
  "packages": "mysql-server-5.6",  
  "id": "2dc7faa0-ffff-4c2b-8cff-bcd949c518a5"  
}
```

5.7.2. List configuration parameters

Method	URI	Description
GET	/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters	Lists the available configuration parameters for a data store version.

Parameter information includes the type, minimum and maximum values, and whether you must restart the instance after you change a parameter value.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.7.2.1. Request

This table shows the URI parameters for the list configuration parameters request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{datastore_version_id}	UUID	The UUID of the data store version.

Example 5.54. List configuration parameters: JSON request

```
GET /v1.0/1234/datastores/versions/692d8418-7a8f-47f1-8060-59846c6e024f/parameters HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.7.2.2. Response

Example 5.55. List configuration parameters: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 8454
Date: Fri, 12 Jun 2015 21:58:13 GMT
```

```
{
  "configuration-parameters": [
    {
      "name": "myisam_sort_buffer_size",
      "min": 4096,
      "max": 18446744073709552000,
      "restart_required": false,
      "type": "integer",
      "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
      "name": "sync_binlog",
```

```

        "min": 0,
        "max": 4294967295,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "name": "max_allowed_packet",
        "min": 1024,
        "max": 1073741824,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "type": "string",
        "name": "character_set_connection",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
        "restart_required": false
    },
    {
        "name": "autocommit",
        "min": 0,
        "max": 1,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "type": "string",
        "name": "character_set_client",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
        "restart_required": false
    },
    {
        "name": "join_buffer_size",
        "min": 128,
        "max": 18446744073709548000,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "name": "local_infile",
        "min": 0,
        "max": 1,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "name": "auto_increment_offset",
        "min": 1,
        "max": 65535,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "name": "max_connections",

```

```

        "min": 1,
        "max": 100000,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "name": "bulk_insert_buffer_size",
        "min": 0,
        "max": 18446744073709552000,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef474dac18-2c6abd35a758"
    },
    {
        "name": "sort_buffer_size",
        "min": 32768,
        "max": 18446744073709552000,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "name": "innodb_log_buffer_size",
        "min": 262144,
        "max": 4294967295,
        "restart_required": true,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "name": "innodb_file_per_table",
        "min": 0,
        "max": 1,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "type": "string",
        "name": "character_set_server",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
        "restart_required": false
    },
    {
        "name": "innodb_buffer_pool_size",
        "min": 5242880,
        "max": 18446744073709552000,
        "restart_required": true,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef474dac18-2c6abd35a758"
    },
    {
        "type": "string",
        "name": "collation_server",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
        "restart_required": false
    },
    {
        "type": "string",

```

```

        "name": "character_set_filesystem",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
        "restart_required": false
    },
    {
        "type": "string",
        "name": "collation_database",
        "datastore_version_id": "f597f709-70ef474d-ac18-2c6abd35a758",
        "restart_required": false
    },
    {
        {
            "name": "innodb_flush_log_at_trx_commit",
            "min": 0,
            "max": 2,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "interactive_timeout",
            "min": 1,
            "max": 65535,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "max_user_connections",
            "min": 0,
            "max": 4294967295,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "innodb_thread_concurrency",
            "min": 0,
            "max": 1000,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "innodb_open_files",
            "min": 10,
            "max": 4294967295,
            "restart_required": true,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "key_buffer_size",
            "min": 8,
            "max": 4294967295,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef474d-ac18-2c6abd35a758"
        },
        {
            "name": "connect_timeout",

```



```

        "min": 2,
        "max": 31536000,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "type": "string",
        "name": "collation_connection",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
        "restart_required": false
    },
    {
        "type": "string",
        "name": "character_set_database",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
        "restart_required": false
    },
    {
        "name": "auto_increment_increment",
        "min": 1,
        "max": 65535,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "name": "max_connect_errors",
        "min": 1,
        "max": 18446744073709552000,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "type": "string",
        "name": "character_set_results",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
        "restart_required": false
    },
    {
        "name": "expire_logs_days",
        "min": 0,
        "max": 99,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "name": "wait_timeout",
        "min": 1,
        "max": 31536000,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "name": "server_id",
        "min": 0,
        "max": 4294967295,

```

```
        "restart_required": false,  
        "type": "integer",  
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"  
    }  
]  
}
```

5.7.3. Show configuration parameter details

Method	URI	Description
GET	/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters/{parameter_name}	Displays details for a configuration parameter associated with a data store version.

Details include the type, minimum and maximum values, and whether you must restart the instance after you change the parameter value.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.7.3.1. Request

This table shows the URI parameters for the show configuration parameter details request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{datastore_version_id}	UUID	The UUID of the data store version.
{parameter_name}	String	The name of the parameter for which to show details.

Example 5.56. Show configuration parameter details: JSON request

```
GET /v1.0/1234/datastores/versions/f8e67741-e767-4137-b394-3fb8a3fafd2f/
parameters/connect_timeout HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.7.3.2. Response

Example 5.57. Show configuration parameter details: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 180
Date: Tue, 16 Jun 2015 21:58:13 GMT
```

```
{
  "name": "connect_timeout",
  "min": 2,
  "max": 31536000,
  "restart_required": false,
  "type": "integer",
  "datastore_version_id": "f8e67741-e767-4137-b394-3fb8a3fafd2f"
}
```

5.8. Configuration groups (configurations)

Creates and lists all configuration groups.

Method	URI	Description
POST	/v1.0/{accountId}/configurations	Creates a configuration group.
GET	/v1.0/{accountId}/configurations	Lists all configuration groups.
DELETE	/v1.0/{accountId}/configurations/{configId}	Deletes a configuration group.
PATCH	/v1.0/{accountId}/configurations/{configId}	Sets new values for a configuration group.
GET	/v1.0/{accountId}/configurations/{configId}	Lists details about a configuration group, including its values.
PUT	/v1.0/{accountId}/configurations/{configId}	Sets new values for a configuration group. Also lets you change the name and description of the configuration group.
GET	/v1.0/{accountId}/configurations/{configId}/instances	Lists the instances associated with the specified configuration group.

5.8.1. Create configuration group

Method	URI	Description
POST	/v1.0/{accountId}/configurations	Creates a configuration group.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.8.1.1. Request

This table shows the URI parameters for the create configuration group request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.

Example 5.58. Create configuration group: JSON request

```
POST /v1.0/1234/configurations HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: d6cafa5b-e0c7-4ab8-948e-7c95f2acd031
Accept: application/json
Content-Type: application/json
```

```
{
  "configuration": {
    "datastore": [
      {
        "type": "mysql"
      }
    ],
    "values": [
      {
        "sync_binlog": 1
      }
    ],
    "name": "group1"
  }
}
```

5.8.1.2. Response

Example 5.59. Create configuration group: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 360
Date: Mon, 6 Jul 2015 21:53:10 GMT
```

```
{
```

```
"updated": "2015-07-01T16:38:27",
"name": "group1",
"created": "2015-07-01T16:38:27",
"instance_count": 0,
"values": {
  "sync_binlog": 1
},
"datastore_version_id": "2dc7faa0-ffff-4c2b-8cff-bcd949c518a5",
"id": "2aa51628-5c42-4086-8682-137caffd2ba6",
"datastore_name": "mysql",
"datastore_version_name": "5.6",
"description": null
}
```

5.8.2. List configuration groups

Method	URI	Description
GET	/v1.0/{accountId}/configurations	Lists all configuration groups.

The list includes the associated data store and data store version for each configuration group.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.8.2.1. Request

This table shows the URI parameters for the list configuration groups request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.

Example 5.60. List configuration groups: JSON request

```
GET /v1.0/1234/configurations HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.8.2.2. Response

Example 5.61. List configuration groups: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 426
Date: Tue, 07 Jul 2012 19:53:04 GMT
```

```
{
  "configurations": [
    {
      "datastore_name": "mysql",
      "updated": "2015-07-01T16:38:27",
      "name": "group1",
      "created": "2015-07-01T16:38:27",
      "datastore_version_name": "5.6",
      "id": "2aa51628-5c42-4086-8682-137caffd2ba6",
      "datastore_version_id": "2dc7faa0-ffff-4c2b-8cff-bcd949c518a5",
      "description": null
    }
  ]
}
```

```
}  
    ]  
}
```


5.8.3. Delete configuration group

Method	URI	Description
DELETE	/v1.0/{accountId}/configurations/{configId}	Deletes a configuration group.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), 422, itemNotFound (404)

5.8.3.1. Request

This table shows the URI parameters for the delete configuration group request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{configId}	String	The ID of the configuration group.

Example 5.62. Delete configuration group: JSON request

```
DELETE /v1.0/1234/configurations/692d8418-7a8f-47f1-8060-59846c6e024f HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.8.4. Patch configuration group

Method	URI	Description
PATCH	/v1.0/{accountId}/configurations/{configId}	Sets new values for a configuration group.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.8.4.1. Request

This table shows the URI parameters for the patch configuration group request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{configId}	String	The ID of the configuration group.

Example 5.63. Patch configuration group: JSON request

```
PATCH /v1.0/1234/configurations/5678 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: d6cafa5b-e0c7-4ab8-948e-7c95f2acd031
Accept: application/json
Content-Type: application/json
```

```
{
  "configuration": {
    "values": {
      "connect_timeout": 17
    }
  }
}
```

5.8.4.2. Response

Example 5.64. Patch configuration group: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 0
Date: Mon, 6 Jul 2015 21:53:10 GMT
```

This operation does not return a response body.

5.8.5. Show configuration group details

Method	URI	Description
GET	/v1.0/{accountId}/configurations/{configId}	Lists details about a configuration group, including its values.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.8.5.1. Request

This table shows the URI parameters for the show configuration group details request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{configId}	String	The ID of the configuration group.

Example 5.65. Show configuration group details: JSON request

```
GET /v1.0/1234/configurations/5678 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.8.5.2. Response

Example 5.66. Show configuration group details: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 138
Date: Fri, 20 Nov 2015 19:53:04 GMT
```

```
{
  "configuration": {
    "datastore_name": "mysql",
    "updated": "2015-11-22T19:07:20",
    "values": {
      "connect_timeout": 17
    },
    "name": "group1",
    "created": "2015-11-20T20:51:24",
    "datastore_version_name": "5.6",
    "instance_count": 1,
    "id": "1c8a4fdd-690c-4e6e-b2e1-148b8d738770",
    "datastore_version_id": "b9f97132-467b-4f8e-b12d-947cfc223ac3",
    "description": null
  }
}
```

```
}  
}
```

5.8.6. Update configuration group

Method	URI	Description
PUT	/v1.0/{accountId}/configurations/{configId}	Sets new values for a configuration group. Also lets you change the name and description of the configuration group.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.8.6.1. Request

This table shows the URI parameters for the update configuration group request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{configId}	String	The ID of the configuration group.

Example 5.67. Update configuration group: JSON request

```
PUT /v1.0/1234/configurations/5678 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: d6cafa5b-e0c7-4ab8-948e-7c95f2acd031
Accept: application/json
Content-Type: application/json
```

```
{
  "configuration": {
    "values": {
      "connect_timeout": 18
    },
    "name": "new_name",
    "description": "New description"
  }
}
```

5.8.6.2. Response

Example 5.68. Update configuration group: JSON response

```
HTTP/1.1 202 OK
Content-Type: application/json
Content-Length: 0
Date: Mon, 6 Jul 2015 21:53:10 GMT
```

This operation does not return a response body.

5.8.7. List configuration group instances

Method	URI	Description
GET	/v1.0/{accountId}/configurations/{configId}/instances	Lists the instances associated with the specified configuration group.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

5.8.7.1. Request

This table shows the URI parameters for the list configuration group instances request:

Name	Type	Description
{accountId}	String	The account ID of the owner of the instance.
{configId}	String	The ID of the configuration group.

Example 5.69. List configuration group instances: JSON request

```
GET /v1.0/1234/configurations/5678/instances HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

5.8.7.2. Response

Example 5.70. List configuration group instances: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 138
Date: Fri, 20 Nov 2015 19:53:04 GMT
```

```
{
  "instances": [
    {
      "id": "7fd2d1d6-a2ef-4a76-8c03-e233db4d86da",
      "name": "master_1"
    }
  ]
}
```

6. Data Processing API v1.1 (CURRENT)

Produce data processing operations.

Method	URI	Description
Plugins		
GET	/v1.1/{tenant_id}/plugins	Lists all registered plugins.
GET	/v1.1/{tenant_id}/plugins/{plugin_name}	Shows details for a plugin.
GET	/v1.1/{tenant_id}/plugins/{plugin_name}/{version}	Shows details for a plugin version.
Image registry		
GET	/v1.1/{tenant_id}/images	Lists all images registered in the registry.
GET	/v1.1/{tenant_id}/images/{image_id}	Shows details for an image.
POST	/v1.1/{tenant_id}/images/{image_id}	Registers an image in the registry.
DELETE	/v1.1/{tenant_id}/images/{image_id}	Removes an image from the registry.
POST	/v1.1/{tenant_id}/images/{image_id}/tag	Adds tags to an image.
POST	/v1.1/{tenant_id}/images/{image_id}/untag	Removes tags from an image.
Node group templates		
GET	/v1.1/{tenant_id}/node-group-templates	Lists available node group templates.
POST	/v1.1/{tenant_id}/node-group-templates	Creates a node group template.
GET	/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}	Shows a node group template, by ID.
DELETE	/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}	Deletes a node group template.
PUT	/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}	Updates a node group template.
Cluster templates		
GET	/v1.1/{tenant_id}/cluster-templates	Lists available cluster templates.
POST	/v1.1/{tenant_id}/cluster-templates	Creates a cluster template.
GET	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Shows details for a cluster template.
PUT	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Updates a cluster template.
DELETE	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Deletes a cluster template.
Clusters		
GET	/v1.1/{tenant_id}/clusters	Lists available clusters.
POST	/v1.1/{tenant_id}/clusters	Creates a cluster.
POST	/v1.1/{tenant_id}/clusters/multiple	Creates multiple clusters.
GET	/v1.1/{tenant_id}/clusters/{cluster_id}	Shows details for a cluster, by ID.

Method	URI	Description
PUT	/v1.1/{tenant_id}/clusters/{cluster_id}	Scales a cluster.
PATCH	/v1.1/{tenant_id}/clusters/{cluster_id}	Updates a cluster.
GET	/v1.1/{tenant_id}/clusters/{cluster_id}	Shows provisioning progress for a cluster.
DELETE	/v1.1/{tenant_id}/clusters/{cluster_id}	Deletes a cluster.
Event log		
GET	/v1.1/{tenant_id}/clusters/{cluster_id}	Shows provisioning progress of cluster.
Data sources		
GET	/v1.1/{tenant_id}/data-sources	Lists all data sources.
POST	/v1.1/{tenant_id}/data-sources	Creates a data source.
GET	/v1.1/{tenant_id}/data-sources/{data_source_id}	Shows details for a data source.
PUT	/v1.1/{tenant_id}/data-sources/{data_source_id}	Updates a data source.
DELETE	/v1.1/{tenant_id}/data-sources/{data_source_id}	Deletes a data source.
Job binary internals		
GET	/v1.1/{tenant_id}/job-binary-internals	Lists the available job binary internals.
PUT	/v1.1/{tenant_id}/job-binary-internals/{name}	Creates a job binary internal.
GET	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}	Shows details for a job binary internal.
DELETE	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}	Deletes a job binary internal.
PATCH	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}	Updates a job binary internal.
GET	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}/data	Shows data for a job binary internal.
Job binaries		
GET	/v1.1/{tenant_id}/job-binaries	Lists the available job binaries.
GET	/v1.1/{tenant_id}/job-binaries	Shows details for a job binary.
POST	/v1.1/{tenant_id}/job-binaries	Creates a job binary.
PUT	/v1.1/{tenant_id}/job-binaries	Updates a job binary.
DELETE	/v1.1/{tenant_id}/job-binaries	Deletes a job binary.
GET	/v1.1/{tenant_id}/job-binaries/{job_binary_id}/data	Shows data for a job binary.
Jobs		
GET	/v1.1/{tenant_id}/jobs	Lists all jobs.
POST	/v1.1/{tenant_id}/jobs	Creates a job object.
GET	/v1.1/{tenant_id}/jobs/{job_id}	Shows details for a job.
DELETE	/v1.1/{tenant_id}/jobs/{job_id}	Removes a job.
PATCH	/v1.1/{tenant_id}/jobs/{job_id}	Updates a job object.
POST	/v1.1/{tenant_id}/jobs/{job_id}/execute	Runs a job.

Method	URI	Description
Job executions		
GET	/v1.1/{tenant_id}/job-executions	Lists available job executions.
GET	/v1.1/{tenant_id}/job-executions/{job_execution_id}	Shows details for a job execution, by ID.
DELETE	/v1.1/{tenant_id}/job-executions/{job_execution_id}	Deletes a job execution.
PATCH	/v1.1/{tenant_id}/job-executions/{job_execution_id}	Updates a job execution.
GET	/v1.1/{tenant_id}/job-executions/{job_execution_id}/refresh-status	Refreshes the status of and shows information for a job execution.
GET	/v1.1/{tenant_id}/job-executions/{job_execution_id}/cancel	Cancels a job execution.
Job types		
GET	/v1.1/{tenant_id}/job-types	Lists all job types.

6.1. Plugins

A plugin object defines the Hadoop or Spark version that it can install and which configurations can be set for the cluster.

Method	URI	Description
GET	/v1.1/{tenant_id}/plugins	Lists all registered plugins.
GET	/v1.1/{tenant_id}/plugins/{plugin_name}	Shows details for a plugin.
GET	/v1.1/{tenant_id}/plugins/{plugin_name}/{version}	Shows details for a plugin version.

6.1.1. List plugins

Method	URI	Description
GET	/v1.1/{tenant_id}/plugins	Lists all registered plugins.

Normal response codes: 200

6.1.1.1. Request

This operation does not accept a request body.

6.1.1.2. Response

Example 6.1. List plugins: JSON response

```
{
  "plugins": [
    {
      "name": "vanilla",
      "description": "The Apache Vanilla plugin provides the ability
to launch upstream Vanilla Apache Hadoop cluster without any management
consoles. It can also deploy the Oozie component.",
      "versions": [
        "1.2.1",
        "2.4.1",
        "2.6.0"
      ],
      "title": "Vanilla Apache Hadoop"
    },
    {
      "name": "hdp",
      "description": "The Hortonworks Sahara plugin automates the
deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
      "versions": [
        "1.3.2",
        "2.0.6"
      ],
      "title": "Hortonworks Data Platform"
    },
    {
      "name": "spark",
      "description": "This plugin provides an ability to launch Spark on
Hadoop CDH cluster without any management consoles.",
      "versions": [
        "1.0.0",
        "0.9.1"
      ],
      "title": "Apache Spark"
    },
    {
      "name": "cdh",
      "description": "The Cloudera Sahara plugin provides the ability to
launch the Cloudera distribution of Apache Hadoop (CDH) with Cloudera Manager
management console.",
      "versions": [
        "5",
```

```
        "5.3.0"  
      ],  
      "title": "Cloudera Plugin"  
    }  
  ]  
}
```

6.1.2. Show plugin details

Method	URI	Description
GET	/v1.1/{tenant_id}/plugins/{plugin_name}	Shows details for a plugin.

Normal response codes: 200

6.1.2.1. Request

This table shows the URI parameters for the show plugin details request:

Name	Type	Description
{plugin_name}	String	Name of the plugin.

This operation does not accept a request body.

6.1.2.2. Response

Example 6.2. Show plugin details: JSON response

```
{
  "plugin": {
    "name": "vanilla",
    "versions": [
      "1.2.1",
      "2.4.1",
      "2.6.0"
    ],
    "title": "Vanilla Apache Hadoop",
    "description": "The Apache Vanilla plugin provides the ability
to launch upstream Vanilla Apache Hadoop cluster without any management
consoles. It can also deploy the Oozie component."
  }
}
```

6.1.3. Show plugin version details

Method	URI	Description
GET	/v1.1/{tenant_id}/plugins/{plugin_name}/{version}	Shows details for a plugin version.

Normal response codes: 200

6.1.3.1. Request

This table shows the URI parameters for the show plugin version details request:

Name	Type	Description
{plugin_name}	String	Name of the plugin.
{version}	String	Version of the plugin.

This operation does not accept a request body.

6.1.3.2. Response

Example 6.3. Show plugin version details: JSON response

```
{
  "plugin": {
    "name": "vanilla",
    "versions": [
      "1.2.1",
      "2.4.1",
      "2.6.0"
    ],
    "description": "The Apache Vanilla plugin provides the ability
to launch upstream Vanilla Apache Hadoop cluster without any management
consoles. It can also deploy the Oozie component.",
    "required_image_tags": [
      "vanilla",
      "2.6.0"
    ],
    "node_processes": {
      "JobFlow": [
        "oozie"
      ],
      "HDFS": [
        "namenode",
        "datanode",
        "secondarynamenode"
      ],
      "YARN": [
        "resourcemanager",
        "nodemanager"
      ],
      "MapReduce": [
        "historyserver"
      ],
      "Hadoop": [],
      "Hive": [
        "hiveserver"
      ]
    }
  }
}
```

```

    },
    "configs": [
      {
        "default_value": "/tmp/hadoop-${user.name}",
        "name": "hadoop.tmp.dir",
        "priority": 2,
        "config_type": "string",
        "applicable_target": "HDFS",
        "is_optional": true,
        "scope": "node",
        "description": "A base for other temporary directories."
      },
      {
        "default_value": true,
        "name": "hadoop.native.lib",
        "priority": 2,
        "config_type": "bool",
        "applicable_target": "HDFS",
        "is_optional": true,
        "scope": "node",
        "description": "Should native hadoop libraries, if present, be
used."
      },
      {
        "default_value": 1024,
        "name": "NodeManager Heap Size",
        "config_values": null,
        "priority": 1,
        "config_type": "int",
        "applicable_target": "YARN",
        "is_optional": false,
        "scope": "node",
        "description": null
      },
      {
        "default_value": true,
        "name": "Enable Swift",
        "config_values": null,
        "priority": 1,
        "config_type": "bool",
        "applicable_target": "general",
        "is_optional": false,
        "scope": "cluster",
        "description": null
      },
      {
        "default_value": true,
        "name": "Enable MySQL",
        "config_values": null,
        "priority": 1,
        "config_type": "bool",
        "applicable_target": "general",
        "is_optional": true,
        "scope": "cluster",
        "description": null
      }
    ],
    "title": "Vanilla Apache Hadoop"
  }
}

```

6.2. Image registry

Use the image registry tool to manage images, add tags to and remove tags from images, and define the user name for an instance operating system. Each plugin lists required tags for an image. To run remote operations, the Data Processing service requires a user name with which to log in to the operating system for an instance.

Method	URI	Description
GET	/v1.1/{tenant_id}/images	Lists all images registered in the registry.
GET	/v1.1/{tenant_id}/images/{image_id}	Shows details for an image.
POST	/v1.1/{tenant_id}/images/{image_id}	Registers an image in the registry.
DELETE	/v1.1/{tenant_id}/images/{image_id}	Removes an image from the registry.
POST	/v1.1/{tenant_id}/images/{image_id}/tag	Adds tags to an image.
POST	/v1.1/{tenant_id}/images/{image_id}/untag	Removes tags from an image.

6.2.1. List images

Method	URI	Description
GET	/v1.1/{tenant_id}/images	Lists all images registered in the registry.

Normal response codes: 200

6.2.1.1. Request

This operation does not accept a request body.

6.2.1.2. Response

Example 6.4. List images: JSON response

```
{
  "images": [
    {
      "name": "ubuntu-vanilla-2.7.1",
      "id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
      "created": "2015-08-06T08:17:14Z",
      "metadata": {
        "_sahara_tag_2.7.1": "True",
        "_sahara_username": "ubuntu",
        "_sahara_tag_vanilla": "True"
      },
      "username": "ubuntu",
      "progress": 100,
      "OS-EXT-IMG-SIZE:size": 998716928,
      "status": "ACTIVE",
      "minDisk": 0,
      "tags": [
        "vanilla",
        "2.7.1"
      ],
      "updated": "2015-09-04T09:35:09Z",
      "minRam": 0,
      "description": null
    },
    {
      "name": "cdh-latest",
      "id": "ff74035b-9da7-4edf-981d-57f270ed337d",
      "created": "2015-09-04T11:56:44Z",
      "metadata": {
        "_sahara_username": "ubuntu",
        "_sahara_tag_5.4.0": "True",
        "_sahara_tag_cdh": "True"
      },
      "username": "ubuntu",
      "progress": 100,
      "OS-EXT-IMG-SIZE:size": 3281453056,
      "status": "ACTIVE",
      "minDisk": 0,
      "tags": [
        "5.4.0",
        "cdh"
      ]
    }
  ]
}
```



```
    ],  
    "updated": "2015-09-04T12:46:42Z",  
    "minRam": 0,  
    "description": null  
  }  
]  
}
```

6.2.2. Show image details

Method	URI	Description
GET	/v1.1/{tenant_id}/images/{image_id}	Shows details for an image.

Normal response codes: 200

6.2.2.1. Request

This table shows the URI parameters for the show image details request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

6.2.2.2. Response

Example 6.5. Show image details: JSON response

```
{
  "image": {
    "updated": "2015-02-03T10:29:32Z",
    "metadata": {
      "_sahara_username": "ubuntu",
      "_sahara_tag_vanilla": "True",
      "_sahara_tag_2.6.0": "True"
    },
    "id": "bb8d12b5-f9bb-49f0-aecb-739b8a9bec89",
    "minDisk": 0,
    "status": "ACTIVE",
    "tags": [
      "vanilla",
      "2.6.0"
    ],
    "minRam": 0,
    "progress": 100,
    "username": "ubuntu",
    "created": "2015-02-03T10:28:39Z",
    "name": "sahara-vanilla-2.6.0-ubuntu-14.04",
    "description": null,
    "OS-EXT-IMG-SIZE:size": 1101856768
  }
}
```

6.2.3. Register image

Method	URI	Description
POST	/v1.1/{tenant_id}/images/{image_id}	Registers an image in the registry.

Normal response codes: 202

6.2.3.1. Request

This table shows the URI parameters for the register image request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

Example 6.6. Register image: JSON request

```
{
  "username": "ubuntu",
  "description": "Ubuntu image for Hadoop 2.7.1"
}
```

6.2.3.2. Response

Example 6.7. Register image: JSON response

```
{
  "image": {
    "updated": "2015-03-24T10:05:10Z",
    "metadata": {
      "_sahara_description": "Ubuntu image for Hadoop 2.7.1",
      "_sahara_username": "ubuntu",
      "_sahara_tag_vanilla": "True",
      "_sahara_tag_2.7.1": "True"
    },
    "id": "bb8d12b5-f9bb-49f0-aecb-739b8a9bec89",
    "minDisk": 0,
    "status": "ACTIVE",
    "tags": [
      "vanilla",
      "2.7.1"
    ],
    "minRam": 0,
    "progress": 100,
    "username": "ubuntu",
    "created": "2015-02-03T10:28:39Z",
    "name": "sahara-vanilla-2.7.1-ubuntu-14.04",
    "description": "Ubuntu image for Hadoop 2.7.1",
    "OS-EXT-IMG-SIZE:size": 1101856768
  }
}
```

6.2.4. Unregister image

Method	URI	Description
DELETE	/v1.1/{tenant_id}/images/{image_id}	Removes an image from the registry.

Normal response codes: 204

6.2.4.1. Request

This table shows the URI parameters for the unregister image request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

6.2.5. Add tags to image

Method	URI	Description
POST	/v1.1/{tenant_id}/images/{image_id}/tag	Adds tags to an image.

Normal response codes: 202

6.2.5.1. Request

This table shows the URI parameters for the add tags to image request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

Example 6.8. Add tags to image: JSON request

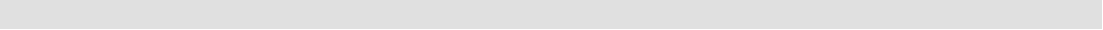
```
{
  "tags": [
    "vanilla",
    "2.7.1",
    "some_other_tag"
  ]
}
```

6.2.5.2. Response

Example 6.9. Add tags to image: JSON response

```
{
  "image": {
    "updated": "2015-03-24T10:18:33Z",
    "metadata": {
      "_sahara_tag_vanilla": "True",
      "_sahara_description": "Ubuntu image for Hadoop 2.7.1",
      "_sahara_username": "ubuntu",
      "_sahara_tag_some_other_tag": "True",
      "_sahara_tag_2.7.1": "True"
    },
    "id": "bb8d12b5-f9bb-49f0-aecb-739b8a9bec89",
    "minDisk": 0,
    "status": "ACTIVE",
    "tags": [
      "vanilla",
      "some_other_tag",
      "2.7.1"
    ],
    "minRam": 0,
    "progress": 100,
    "username": "ubuntu",
    "created": "2015-02-03T10:28:39Z",
    "name": "sahara-vanilla-2.6.0-ubuntu-14.04",
    "description": "Ubuntu image for Hadoop 2.7.1",
    "OS-EXT-IMG-SIZE:size": 1101856768
  }
}
```

```
}
```



6.2.6. Remove tags from image

Method	URI	Description
POST	/v1.1/{tenant_id}/images/{image_id}/untag	Removes tags from an image.

Normal response codes: 202

6.2.6.1. Request

This table shows the URI parameters for the remove tags from image request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

Example 6.10. Remove tags from image: JSON request

```
{
  "tags": [
    "some_other_tag"
  ]
}
```

6.2.6.2. Response

Example 6.11. Remove tags from image: JSON response

```
{
  "image": {
    "updated": "2015-03-24T10:18:33Z",
    "metadata": {
      "_sahara_tag_vanilla": "True",
      "_sahara_description": "Ubuntu image for Hadoop 2.7.1",
      "_sahara_username": "ubuntu",
      "_sahara_tag_some_other_tag": "True",
      "_sahara_tag_2.7.1": "True"
    },
    "id": "bb8d12b5-f9bb-49f0-aecb-739b8a9bec89",
    "minDisk": 0,
    "status": "ACTIVE",
    "tags": [
      "vanilla",
      "some_other_tag",
      "2.7.1"
    ],
    "minRam": 0,
    "progress": 100,
    "username": "ubuntu",
    "created": "2015-02-03T10:28:39Z",
    "name": "sahara-vanilla-2.6.0-ubuntu-14.04",
    "description": "Ubuntu image for Hadoop 2.7.1",
    "OS-EXT-IMG-SIZE:size": 1101856768
  }
}
```

6.3. Node group templates

A cluster is a group of nodes with the same configuration. A node group template configures a node in the cluster.

A template configures Hadoop processes and VM characteristics, such as the number of reduced slots for task tracker, the number of CPUs, and the amount of RAM. The template specifies the VM characteristics through an OpenStack flavor.

Method	URI	Description
GET	/v1.1/{tenant_id}/node-group-templates	Lists available node group templates.
POST	/v1.1/{tenant_id}/node-group-templates	Creates a node group template.
GET	/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}	Shows a node group template, by ID.
DELETE	/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}	Deletes a node group template.
PUT	/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}	Updates a node group template.

6.3.1. List node group templates

Method	URI	Description
GET	/v1.1/{tenant_id}/node-group-templates	Lists available node group templates.

Normal response codes: 200

6.3.1.1. Request

This operation does not accept a request body.

6.3.1.2. Response

Example 6.12. List node group templates: JSON response

```
{
  "node_group_templates": [
    {
      "is_public": false,
      "image_id": null,
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "shares": null,
      "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
      "node_configs": {},
      "auto_security_group": false,
      "is_default": false,
      "availability_zone": null,
      "plugin_name": "vanilla",
      "flavor_id": "2",
      "id": "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
      "description": null,
      "hadoop_version": "2.7.1",
      "use_autoconfig": true,
      "volumes_availability_zone": null,
      "created_at": "2015-09-14T10:20:11",
      "is_protected": false,
      "updated_at": null,
      "volumes_per_node": 0,
      "is_proxy_gateway": false,
      "name": "master",
      "volume_mount_prefix": "/volumes/disk",
      "node_processes": [
        "namenode",
        "resourcemanager",
        "oozie",
        "historyserver"
      ],
      "volumes_size": 0,
      "volume_local_to_instance": false,
      "security_groups": null,
      "volume_type": null
    },
    {
      "is_public": false,
      "image_id": null,
```

```
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "shares": null,
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
    "node_configs": {},
    "auto_security_group": false,
    "is_default": false,
    "availability_zone": null,
    "plugin_name": "vanilla",
    "flavor_id": "2",
    "id": "846edb31-add5-46e6-a4ee-a4c339f99251",
    "description": null,
    "hadoop_version": "2.7.1",
    "use_autoconfig": true,
    "volumes_availability_zone": null,
    "created_at": "2015-09-14T10:27:00",
    "is_protected": false,
    "updated_at": null,
    "volumes_per_node": 0,
    "is_proxy_gateway": false,
    "name": "worker",
    "volume_mount_prefix": "/volumes/disk",
    "node_processes": [
        "datanode",
        "nodemanager"
    ],
    "volumes_size": 0,
    "volume_local_to_instance": false,
    "security_groups": null,
    "volume_type": null
  }
]
```

6.3.2. Create node group template

Method	URI	Description
POST	/v1.1/{tenant_id}/node-group-templates	Creates a node group template.

Normal response codes: 202

6.3.2.1. Request

Example 6.13. Create node group template: JSON request

```
{
  "plugin_name": "vanilla",
  "hadoop_version": "2.7.1",
  "node_processes": [
    "namenode",
    "resourcemanager",
    "oozie",
    "historyserver"
  ],
  "name": "master",
  "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
  "flavor_id": "2"
}
```

6.3.2.2. Response

Example 6.14. Create node group template: JSON response

```
{
  "node_group_template": {
    "is_public": false,
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
    "node_configs": {},
    "auto_security_group": false,
    "is_default": false,
    "availability_zone": null,
    "plugin_name": "vanilla",
    "is_protected": false,
    "flavor_id": "2",
    "id": "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
    "hadoop_version": "2.7.1",
    "use_autoconfig": true,
    "volumes_availability_zone": null,
    "created_at": "2015-09-14T10:20:11",
    "security_groups": null,
    "volumes_per_node": 0,
    "is_proxy_gateway": false,
    "name": "master",
    "volume_mount_prefix": "/volumes/disk",
    "node_processes": [
      "namenode",
      "resourcemanager",
      "oozie",
    ]
  }
}
```

```
        "historyserver"  
      ],  
      "volumes_size": 0,  
      "volume_local_to_instance": false,  
      "volume_type": null  
    }  
  }
```

6.3.3. Show node group template details

Method	URI	Description
GET	/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}	Shows a node group template, by ID.

Normal response codes: 200

6.3.3.1. Request

This table shows the URI parameters for the show node group template details request:

Name	Type	Description
{node_group_template_id}	UUID	The UUID of the node group template.

This operation does not accept a request body.

6.3.3.2. Response

Example 6.15. Show node group template details: JSON response

```
{
  "node_group_template": {
    "is_public": false,
    "image_id": null,
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "shares": null,
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
    "node_configs": {},
    "auto_security_group": false,
    "is_default": false,
    "availability_zone": null,
    "plugin_name": "vanilla",
    "flavor_id": "2",
    "id": "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
    "description": null,
    "hadoop_version": "2.7.1",
    "use_autoconfig": true,
    "volumes_availability_zone": null,
    "created_at": "2015-09-14T10:20:11",
    "is_protected": false,
    "updated_at": null,
    "volumes_per_node": 0,
    "is_proxy_gateway": false,
    "name": "master",
    "volume_mount_prefix": "/volumes/disk",
    "node_processes": [
      "namenode",
      "resourcemanager",
      "oozie",
      "historyserver"
    ],
    "volumes_size": 0,
    "volume_local_to_instance": false,
    "security_groups": null,
  }
}
```

```
    "volume_type": null  
  }  
}
```

6.3.4. Delete node group template

Method	URI	Description
DELETE	/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}	Deletes a node group template.

Normal response codes: 204

6.3.4.1. Request

This table shows the URI parameters for the delete node group template request:

Name	Type	Description
{node_group_template_id}	UUID	The UUID of the node group template.

This operation does not accept a request body.

6.3.5. Update node group template

Method	URI	Description
PUT	/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}	Updates a node group template.

Normal response codes: 202

6.3.5.1. Request

This table shows the URI parameters for the update node group template request:

Name	Type	Description
{node_group_template_id}	UUID	The UUID of the node group template.

Example 6.16. Update node group template: JSON request

```
{
  "plugin_name": "vanilla",
  "hadoop_version": "2.7.1",
  "node_processes": [
    "datanode"
  ],
  "name": "new",
  "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
  "flavor_id": "2"
}
```

6.3.5.2. Response

Example 6.17. Update node group template: JSON response

```
{
  "node_group_template": {
    "is_public": false,
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
    "node_configs": {},
    "auto_security_group": false,
    "is_default": false,
    "availability_zone": null,
    "plugin_name": "vanilla",
    "is_protected": false,
    "flavor_id": "2",
    "id": "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
    "hadoop_version": "2.7.1",
    "use_autoconfig": true,
    "volumes_availability_zone": null,
    "created_at": "2015-09-14T10:20:11",
    "security_groups": null,
    "volumes_per_node": 0,
    "is_proxy_gateway": false,
    "name": "new",
    "volume_mount_prefix": "/volumes/disk",
    "node_processes": [
```



```
        "datanode"  
      ],  
      "volumes_size": 0,  
      "volume_local_to_instance": false,  
      "volume_type": null  
    }  
  }  
}
```

6.4. Cluster templates

A cluster template configures a Hadoop cluster. A cluster template lists node groups with the number of instances in each group. You can also define cluster-scoped configurations in a cluster template.

Method	URI	Description
GET	/v1.1/{tenant_id}/cluster-templates	Lists available cluster templates.
POST	/v1.1/{tenant_id}/cluster-templates	Creates a cluster template.
GET	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Shows details for a cluster template.
PUT	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Updates a cluster template.
DELETE	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Deletes a cluster template.

6.4.1. List cluster templates

Method	URI	Description
GET	/v1.1/{tenant_id}/cluster-templates	Lists available cluster templates.

Normal response codes: 200

6.4.1.1. Request

This operation does not accept a request body.

6.4.1.2. Response

Example 6.18. List cluster templates: JSON response

```
{
  "cluster_templates": [
    {
      "is_public": false,
      "anti_affinity": [],
      "name": "cluster-template",
      "created_at": "2015-09-14T10:38:44",
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "cluster_configs": {},
      "shares": null,
      "id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
      "default_image_id": null,
      "is_default": false,
      "updated_at": null,
      "plugin_name": "vanilla",
      "node_groups": [
        {
          "image_id": null,
          "shares": null,
          "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
          "node_configs": {},
          "auto_security_group": false,
          "availability_zone": null,
          "count": 1,
          "flavor_id": "2",
          "id": "1751c04e-8f39-467e-a421-480961172d4b",
          "security_groups": null,
          "use_autoconfig": true,
          "volumes_availability_zone": null,
          "created_at": "2015-09-14T10:38:44",
          "node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
          "updated_at": null,
          "volumes_per_node": 0,
          "is_proxy_gateway": false,
          "name": "master",
          "volume_mount_prefix": "/volumes/disk",
          "node_processes": [
            "namenode",
            "resourcemanager",
```

```

        "oozie",
        "historyserver"
    ],
    "volumes_size": 0,
    "volume_local_to_instance": false,
    "volume_type": null
},
{
    "image_id": null,
    "shares": null,
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
    "node_configs": {},
    "auto_security_group": false,
    "availability_zone": null,
    "count": 3,
    "flavor_id": "2",
    "id": "3ee85068-c455-4391-9db2-b54a20b99df3",
    "security_groups": null,
    "use_autoconfig": true,
    "volumes_availability_zone": null,
    "created_at": "2015-09-14T10:38:44",
    "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
    "updated_at": null,
    "volumes_per_node": 0,
    "is_proxy_gateway": false,
    "name": "worker",
    "volume_mount_prefix": "/volumes/disk",
    "node_processes": [
        "datanode",
        "nodemanager"
    ],
    "volumes_size": 0,
    "volume_local_to_instance": false,
    "volume_type": null
}
],
    "neutron_management_network": "b1610452-2933-46b0-
bf31-660cfa5621bd",
    "hadoop_version": "2.7.1",
    "use_autoconfig": true,
    "description": null,
    "is_protected": false
},
{
    "is_public": true,
    "anti_affinity": [],
    "name": "asd",
    "created_at": "2015-08-18T08:39:39",
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "cluster_configs": {
        "general": {}
    },
    "shares": null,
    "id": "5a9c787c-2078-4f7d-9a66-27759be9051b",
    "default_image_id": null,
    "is_default": false,
    "updated_at": "2015-09-14T08:41:15",
    "plugin_name": "vanilla",

```

```

        "node_groups": [
            {
                "image_id": null,
                "shares": null,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
                "node_configs": {},
                "auto_security_group": true,
                "availability_zone": "",
                "count": 1,
                "flavor_id": "2",
                "id": "a65864dd-3f99-4d29-a011-f7711cc23fa0",
                "security_groups": [],
                "use_autoconfig": true,
                "volumes_availability_zone": null,
                "created_at": "2015-08-18T08:39:39",
                "node_group_template_id":
"42ce49de-1b8f-41d5-8f4a-244ec0826d92",
                "updated_at": null,
                "volumes_per_node": 1,
                "is_proxy_gateway": false,
                "name": "asd",
                "volume_mount_prefix": "/volumes/disk",
                "node_processes": [
                    "namenode",
                    "jobtracker"
                ],
                "volumes_size": 10,
                "volume_local_to_instance": false,
                "volume_type": null
            }
        ],
        "neutron_management_network": null,
        "hadoop_version": "2.7.1",
        "use_autoconfig": true,
        "description": "",
        "is_protected": false
    }
}

```

6.4.2. Create cluster templates

Method	URI	Description
POST	/v1.1/{tenant_id}/cluster-templates	Creates a cluster template.

Normal response codes: 202

6.4.2.1. Request

Example 6.19. Create cluster template: JSON request

```
{
  "plugin_name": "vanilla",
  "hadoop_version": "2.7.1",
  "node_groups": [
    {
      "name": "worker",
      "count": 3,
      "node_group_template_id": "846edb31-add5-46e6-a4ee-a4c339f99251"
    },
    {
      "name": "master",
      "count": 1,
      "node_group_template_id": "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae"
    }
  ],
  "name": "cluster-template"
}
```

6.4.2.2. Response

Example 6.20. Create cluster template: JSON response

```
{
  "cluster_template": {
    "is_public": false,
    "anti_affinity": [],
    "name": "cluster-template",
    "created_at": "2015-09-14T10:38:44",
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "cluster_configs": {},
    "shares": null,
    "id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
    "default_image_id": null,
    "is_default": false,
    "updated_at": null,
    "plugin_name": "vanilla",
    "node_groups": [
      {
        "image_id": null,
        "shares": null,
        "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
        "node_configs": {},
        "auto_security_group": false,
        "availability_zone": null,
        "count": 1,

```

```

        "flavor_id": "2",
        "id": "1751c04e-8f39-467e-a421-480961172d4b",
        "security_groups": null,
        "use_autoconfig": true,
        "volumes_availability_zone": null,
        "created_at": "2015-09-14T10:38:44",
        "node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
        "updated_at": null,
        "volumes_per_node": 0,
        "is_proxy_gateway": false,
        "name": "master",
        "volume_mount_prefix": "/volumes/disk",
        "node_processes": [
            "namenode",
            "resourcemanager",
            "oozie",
            "historyserver"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "volume_type": null
    },
    {
        "image_id": null,
        "shares": null,
        "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
        "node_configs": {},
        "auto_security_group": false,
        "availability_zone": null,
        "count": 3,
        "flavor_id": "2",
        "id": "3ee85068-c455-4391-9db2-b54a20b99df3",
        "security_groups": null,
        "use_autoconfig": true,
        "volumes_availability_zone": null,
        "created_at": "2015-09-14T10:38:44",
        "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
        "updated_at": null,
        "volumes_per_node": 0,
        "is_proxy_gateway": false,
        "name": "worker",
        "volume_mount_prefix": "/volumes/disk",
        "node_processes": [
            "datanode",
            "nodemanager"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "volume_type": null
    }
],
"neutron_management_network": null,
"hadoop_version": "2.7.1",
"use_autoconfig": true,
"description": null,
"is_protected": false
}

```

6.4.3. Show cluster template details

Method	URI	Description
GET	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Shows details for a cluster template.

Normal response codes: 200

6.4.3.1. Request

This table shows the URI parameters for the show cluster template details request:

Name	Type	Description
{cluster_template_id}	UUID	The unique identifier of the cluster template.

This operation does not accept a request body.

6.4.3.2. Response

Example 6.21. Show cluster template details: JSON response

```
{
  "cluster_templates": [
    {
      "is_public": false,
      "anti_affinity": [],
      "name": "cluster-template",
      "created_at": "2015-09-14T10:38:44",
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "cluster_configs": {},
      "shares": null,
      "id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
      "default_image_id": null,
      "is_default": false,
      "updated_at": null,
      "plugin_name": "vanilla",
      "node_groups": [
        {
          "image_id": null,
          "shares": null,
          "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
          "node_configs": {},
          "auto_security_group": false,
          "availability_zone": null,
          "count": 1,
          "flavor_id": "2",
          "id": "1751c04e-8f39-467e-a421-480961172d4b",
          "security_groups": null,
          "use_autoconfig": true,
          "volumes_availability_zone": null,
          "created_at": "2015-09-14T10:38:44",
          "node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
          "updated_at": null,
          "volumes_per_node": 0,
          "is_proxy_gateway": false,

```

```

        "name": "master",
        "volume_mount_prefix": "/volumes/disk",
        "node_processes": [
            "namenode",
            "resourcemanager",
            "oozie",
            "historyserver"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "volume_type": null
    },
    {
        "image_id": null,
        "shares": null,
        "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
        "node_configs": {},
        "auto_security_group": false,
        "availability_zone": null,
        "count": 3,
        "flavor_id": "2",
        "id": "3ee85068-c455-4391-9db2-b54a20b99df3",
        "security_groups": null,
        "use_autoconfig": true,
        "volumes_availability_zone": null,
        "created_at": "2015-09-14T10:38:44",
        "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
        "updated_at": null,
        "volumes_per_node": 0,
        "is_proxy_gateway": false,
        "name": "worker",
        "volume_mount_prefix": "/volumes/disk",
        "node_processes": [
            "datanode",
            "nodemanager"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "volume_type": null
    }
],
    "neutron_management_network": "b1610452-2933-46b0-
bf31-660cfa5621bd",
    "hadoop_version": "2.7.1",
    "use_autoconfig": true,
    "description": null,
    "is_protected": false
},
{
    "is_public": true,
    "anti_affinity": [],
    "name": "asd",
    "created_at": "2015-08-18T08:39:39",
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "cluster_configs": {
        "general": {}
    },
    "shares": null,

```



```

        "id": "5a9c787c-2078-4f7d-9a66-27759be9051b",
        "default_image_id": null,
        "is_default": false,
        "updated_at": "2015-09-14T08:41:15",
        "plugin_name": "vanilla",
        "node_groups": [
            {
                "image_id": null,
                "shares": null,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
                "node_configs": {},
                "auto_security_group": true,
                "availability_zone": "",
                "count": 1,
                "flavor_id": "2",
                "id": "a65864dd-3f99-4d29-a011-f7711cc23fa0",
                "security_groups": [],
                "use_autoconfig": true,
                "volumes_availability_zone": null,
                "created_at": "2015-08-18T08:39:39",
                "node_group_template_id":
"42ce49de-1b8f-41d5-8f4a-244ec0826d92",
                "updated_at": null,
                "volumes_per_node": 1,
                "is_proxy_gateway": false,
                "name": "asd",
                "volume_mount_prefix": "/volumes/disk",
                "node_processes": [
                    "namenode",
                    "jobtracker"
                ],
                "volumes_size": 10,
                "volume_local_to_instance": false,
                "volume_type": null
            }
        ],
        "neutron_management_network": null,
        "hadoop_version": "2.7.1",
        "use_autoconfig": true,
        "description": "",
        "is_protected": false
    }
]
}

```

6.4.4. Update cluster templates

Method	URI	Description
PUT	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Updates a cluster template.

Normal response codes: 202

6.4.4.1. Request

This table shows the URI parameters for the update cluster templates request:

Name	Type	Description
{cluster_template_id}	UUID	The unique identifier of the cluster template.

Example 6.22. Update the cluster template: JSON request

```
{
  "description": "Updated template",
  "plugin_name": "vanilla",
  "hadoop_version": "2.7.1",
  "name": "vanilla-updated",
  "cluster_configs": {
    "HDFS": {
      "dfs.replication": 2
    }
  }
}
```

6.4.4.2. Response

Example 6.23. Update cluster templates: JSON response

```
{
  "cluster_template": {
    "is_public": false,
    "anti_affinity": [],
    "name": "vanilla-updated",
    "created_at": "2015-08-21T08:41:24",
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "cluster_configs": {
      "HDFS": {
        "dfs.replication": 2
      }
    },
    "shares": null,
    "id": "84d47e85-6094-473f-bf6d-5a7e6e86564e",
    "default_image_id": null,
    "is_default": false,
    "updated_at": "2015-09-14T10:45:57",
    "plugin_name": "vanilla",
    "node_groups": [
      {
        "image_id": null,
        "shares": null,

```

```

        "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
        "node_configs": {
            "YARN": {},
            "JobFlow": {},
            "MapReduce": {},
            "Hive": {},
            "Hadoop": {},
            "HDFS": {}
        },
        "auto_security_group": true,
        "availability_zone": "",
        "count": 1,
        "flavor_id": "3",
        "id": "57b966ab-617e-4735-bf60-0cb991208a52",
        "security_groups": [],
        "use_autoconfig": true,
        "volumes_availability_zone": null,
        "created_at": "2015-08-21T08:41:24",
        "node_group_template_id": "a5533187-3f14-42c3-
ba3a-196c13fe0fb5",
        "updated_at": null,
        "volumes_per_node": 0,
        "is_proxy_gateway": false,
        "name": "all",
        "volume_mount_prefix": "/volumes/disk",
        "node_processes": [
            "namenode",
            "datanode",
            "historyserver",
            "resourcemanager",
            "nodemanager",
            "oozie"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "volume_type": null
    }
},
"neutron_management_network": null,
"hadoop_version": "2.7.1",
"use_autoconfig": true,
"description": "Updated template",
"is_protected": false
}
}

```

6.4.5. Delete cluster template

Method	URI	Description
DELETE	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Deletes a cluster template.

Normal response codes: 204

6.4.5.1. Request

This table shows the URI parameters for the delete cluster template request:

Name	Type	Description
{cluster_template_id}	UUID	The unique identifier of the cluster template.

This operation does not accept a request body.

6.5. Clusters

A cluster is a group of nodes with the same configuration.

Method	URI	Description
GET	/v1.1/{tenant_id}/clusters	Lists available clusters.
POST	/v1.1/{tenant_id}/clusters	Creates a cluster.
POST	/v1.1/{tenant_id}/clusters/multiple	Creates multiple clusters.
GET	/v1.1/{tenant_id}/clusters/{cluster_id}	Shows details for a cluster, by ID.
PUT	/v1.1/{tenant_id}/clusters/{cluster_id}	Scales a cluster.
PATCH	/v1.1/{tenant_id}/clusters/{cluster_id}	Updates a cluster.
GET	/v1.1/{tenant_id}/clusters/{cluster_id}	Shows provisioning progress for a cluster.
DELETE	/v1.1/{tenant_id}/clusters/{cluster_id}	Deletes a cluster.

6.5.1. List available clusters

Method	URI	Description
GET	/v1.1/{tenant_id}/clusters	Lists available clusters.

Normal response codes: 200

6.5.1.1. Request

This operation does not accept a request body.

6.5.1.2. Response

Example 6.24. List available clusters: JSON response

```
{
  "clusters": [
    {
      "is_public": false,
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "shares": null,
      "status_description": "",
      "plugin_name": "vanilla",
      "neutron_management_network": "b1610452-2933-46b0-bf31-660cfa5621bd",
      "info": {
        "YARN": {
          "Web UI": "http://172.18.168.115:8088",
          "ResourceManager": "http://172.18.168.115:8032"
        },
        "HDFS": {
          "Web UI": "http://172.18.168.115:50070",
          "NameNode": "hdfs://vanilla-cluster-master-0:9000"
        },
        "JobFlow": {
          "Oozie": "http://172.18.168.115:11000"
        },
        "MapReduce JobHistory Server": {
          "Web UI": "http://172.18.168.115:19888"
        }
      },
      "user_keypair_id": "apavlov",
      "management_public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCFe9AR0
+t9CybtuC1+cusDTeQL7wos1+U2dKPLCUJvNU0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamck8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rj0wAQANhL7KatGLyCPQg9fq0oaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZkTOvEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9Ph0SqcrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpdtFLH4eN5MFDh Generated-by-Sahara\n",
      "id": "e172d86c-906d-418e-a29c-6189f53bfa42",
      "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
      "node_groups": [
        {
          "image_id": null,
          "shares": null,

```

```

    "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
    "node_configs": {
      "YARN": {
        "yarn.nodemanager.vmem-check-enabled": "false",
        "yarn.scheduler.maximum-allocation-mb": 2048,
        "yarn.scheduler.minimum-allocation-mb": 256,
        "yarn.nodemanager.resource.memory-mb": 2048
      },
      "MapReduce": {
        "yarn.app.mapreduce.am.resource.mb": 256,
        "mapreduce.task.io.sort.mb": 102,
        "mapreduce.reduce.java.opts": "-Xmx409m",
        "mapreduce.reduce.memory.mb": 512,
        "mapreduce.map.memory.mb": 256,
        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
        "mapreduce.map.java.opts": "-Xmx204m"
      }
    },
    "auto_security_group": false,
    "availability_zone": null,
    "count": 1,
    "flavor_id": "2",
    "id": "0fe07f2a-0275-4bc0-93b2-c3c1e48e2815",
    "security_groups": null,
    "use_autoconfig": true,
    "instances": [
      {
        "created_at": "2015-09-14T10:57:36",
        "id": "4867d92e-cc7b-4cde-9a1a-149e91caa491",
        "management_ip": "172.18.168.115",
        "updated_at": "2015-09-14T10:57:39",
        "instance_id":
"b9f16a07-88fc-423e-83a3-489598fe6737",
        "internal_ip": "10.50.0.60",
        "instance_name": "vanilla-cluster-master-0"
      }
    ],
    "volumes_availability_zone": null,
    "created_at": "2015-09-14T10:57:11",
    "node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
    "updated_at": "2015-09-14T10:57:36",
    "volumes_per_node": 0,
    "is_proxy_gateway": false,
    "name": "master",
    "volume_mount_prefix": "/volumes/disk",
    "node_processes": [
      "namenode",
      "resourcemanager",
      "oozie",
      "historyserver"
    ],
    "volumes_size": 0,
    "volume_local_to_instance": false,
    "volume_type": null
  },
  {
    "image_id": null,
    "shares": null,

```

```

    "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
    "node_configs": {
      "YARN": {
        "yarn.nodemanager.vmem-check-enabled": "false",
        "yarn.scheduler.maximum-allocation-mb": 2048,
        "yarn.scheduler.minimum-allocation-mb": 256,
        "yarn.nodemanager.resource.memory-mb": 2048
      },
      "MapReduce": {
        "yarn.app.mapreduce.am.resource.mb": 256,
        "mapreduce.task.io.sort.mb": 102,
        "mapreduce.reduce.java.opts": "-Xmx409m",
        "mapreduce.reduce.memory.mb": 512,
        "mapreduce.map.memory.mb": 256,
        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
        "mapreduce.map.java.opts": "-Xmx204m"
      }
    },
    "auto_security_group": false,
    "availability_zone": null,
    "count": 3,
    "flavor_id": "2",
    "id": "c7a3bea4-c898-446b-8c67-6d378d4c06c4",
    "security_groups": null,
    "use_autoconfig": true,
    "instances": [
      {
        "created_at": "2015-09-14T10:57:37",
        "id": "f3633b30-c1e4-4144-930b-ab5b780b87be",
        "management_ip": "172.18.168.118",
        "updated_at": "2015-09-14T10:57:39",
        "instance_id": "0cf1ee81-aa72-48da-
be2c-65bc2fa51f8f",
        "internal_ip": "10.50.0.63",
        "instance_name": "vanilla-cluster-worker-0"
      },
      {
        "created_at": "2015-09-14T10:57:37",
        "id": "0d66fd93-f277-4a94-b46a-f5866aa0c38f",
        "management_ip": "172.18.168.117",
        "updated_at": "2015-09-14T10:57:40",
        "instance_id": "4a937391-
b594-4ad0-9a53-00a99a691383",
        "internal_ip": "10.50.0.62",
        "instance_name": "vanilla-cluster-worker-1"
      },
      {
        "created_at": "2015-09-14T10:57:37",
        "id": "0982cefd-5c58-436e-8f1e-c1d0830f18a7",
        "management_ip": "172.18.168.116",
        "updated_at": "2015-09-14T10:57:40",
        "instance_id":
"839b1d56-6d0d-4aa4-9d05-30e029c276f8",
        "internal_ip": "10.50.0.61",
        "instance_name": "vanilla-cluster-worker-2"
      }
    ],
    "volumes_availability_zone": null,
    "created_at": "2015-09-14T10:57:11",

```

```

a4c339f99251",
    "node_group_template_id": "846edb31-add5-46e6-a4ee-
    "updated_at": "2015-09-14T10:57:37",
    "volumes_per_node": 0,
    "is_proxy_gateway": false,
    "name": "worker",
    "volume_mount_prefix": "/volumes/disk",
    "node_processes": [
        "datanode",
        "nodemanager"
    ],
    "volumes_size": 0,
    "volume_local_to_instance": false,
    "volume_type": null
  },
  "provision_progress": [
    {
      "created_at": "2015-09-14T10:57:18",
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "id": "0a6d95f9-30f4-4434-823a-a38a7999a5af",
      "step_type": "Engine: create cluster",
      "step_name": "Create Heat stack",
      "updated_at": "2015-09-14T10:57:38",
      "successful": true,
      "total": 1,
      "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    },
    {
      "created_at": "2015-09-14T10:58:16",
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "id": "29f2b587-c34c-4871-9ed9-9235b411cd9a",
      "step_type": "Engine: create cluster",
      "step_name": "Configure instances",
      "updated_at": "2015-09-14T10:58:22",
      "successful": true,
      "total": 4,
      "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    },
    {
      "created_at": "2015-09-14T11:00:27",
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "id": "36f1efde-90f9-41c1-b409-aa1cf9623e3e",
      "step_type": "Plugin: start cluster",
      "step_name": "Start the following process(es): Oozie",
      "updated_at": "2015-09-14T11:01:15",
      "successful": true,
      "total": 1,
      "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    },
    {
      "created_at": "2015-09-14T10:58:22",
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "id": "602bcc27-3a2d-42c8-8aca-ebc475319c72",
      "step_type": "Plugin: configure cluster",
      "step_name": "Configure instances",
      "updated_at": "2015-09-14T10:59:21",
      "successful": true,
      "total": 4,
      "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    }
  ]
}

```



```

    },
    {
      "created_at": "2015-09-14T10:59:21",
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "id": "7e291df1-2d32-410d-ae89-33ab6f83cf17",
      "step_type": "Plugin: configure cluster",
      "step_name": "Configure topology data",
      "updated_at": "2015-09-14T10:59:37",
      "successful": true,
      "total": 1,
      "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    },
    {
      "created_at": "2015-09-14T11:00:01",
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "id": "8ab7933c-ad61-4a4f-88db-23ce78ee10f6",
      "step_type": "Plugin: start cluster",
      "step_name": "Start the following process(es): DataNodes,
NodeManagers",
      "updated_at": "2015-09-14T11:00:11",
      "successful": true,
      "total": 3,
      "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    },
    {
      "created_at": "2015-09-14T11:00:11",
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "id": "9c8dc016-8c5b-4e80-9857-80c41f6bd971",
      "step_type": "Plugin: start cluster",
      "step_name": "Await DataNodes start up",
      "updated_at": "2015-09-14T11:00:21",
      "successful": true,
      "total": 1,
      "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    },
    {
      "created_at": "2015-09-14T11:00:21",
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "id": "c6327532-222b-416c-858f-73dbb32b8e97",
      "step_type": "Plugin: start cluster",
      "step_name": "Start the following process(es):
HistoryServer",
      "updated_at": "2015-09-14T11:00:27",
      "successful": true,
      "total": 1,
      "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    },
    {
      "created_at": "2015-09-14T10:57:41",
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "id": "d3eca726-8b44-473a-ac29-fba45a893725",
      "step_type": "Engine: create cluster",
      "step_name": "Wait for instance accessibility",
      "updated_at": "2015-09-14T10:58:14",
      "successful": true,
      "total": 4,
      "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    },
    {
      "created_at": "2015-09-14T10:58:14",

```

```

        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "id": "d7a875ff-64bf-41aa-882d-b5061c8ee152",
        "step_type": "Engine: create cluster",
        "step_name": "Mount volumes to instances",
        "updated_at": "2015-09-14T10:58:15",
        "successful": true,
        "total": 0,
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    },
    {
        "created_at": "2015-09-14T10:59:55",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "id": "ded7d227-10b8-4cb0-ab6c-25da1462bb7a",
        "step_type": "Plugin: start cluster",
        "step_name": "Start the following process(es):
ResourceManager",
        "updated_at": "2015-09-14T11:00:00",
        "successful": true,
        "total": 1,
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    },
    {
        "created_at": "2015-09-14T10:59:38",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "id": "e1701ff5-930a-4212-945a-43515dfe24d1",
        "step_type": "Plugin: start cluster",
        "step_name": "Start the following process(es): NameNode",
        "updated_at": "2015-09-14T10:59:54",
        "successful": true,
        "total": 1,
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    },
    {
        "created_at": "2015-09-14T10:57:38",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "id": "eaf0ab1b-bf8f-48f0-8f2c-fa4f82f539b9",
        "step_type": "Engine: create cluster",
        "step_name": "Assign IPs",
        "updated_at": "2015-09-14T10:57:41",
        "successful": true,
        "total": 4,
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
    }
],
"hadoop_version": "2.7.1",
"use_autoconfig": true,
"trust_id": null,
"description": null,
"created_at": "2015-09-14T10:57:11",
"is_protected": false,
"updated_at": "2015-09-14T11:01:15",
"is_transient": false,
"cluster_configs": {
    "HDFS": {
        "dfs.replication": 3
    }
},
"anti_affinity": [],
"name": "vanilla-cluster",
"default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",

```

```
        "status": "Active"  
      }  
    ]  
  }
```

6.5.2. Create cluster

Method	URI	Description
POST	/v1.1/{tenant_id}/clusters	Creates a cluster.

Normal response codes: 202

6.5.2.1. Request

Example 6.25. Create cluster: JSON request

```
{
  "plugin_name": "vanilla",
  "hadoop_version": "2.7.1",
  "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
  "default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
  "user_keypair_id": "test",
  "name": "vanilla-cluster",
  "neutron_management_network": "b1610452-2933-46b0-bf31-660cfa5621bd"
}
```

6.5.2.2. Response

Example 6.26. Create cluster: JSON response

```
{
  "cluster": {
    "is_public": false,
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "shares": null,
    "status_description": "",
    "plugin_name": "vanilla",
    "neutron_management_network": "b1610452-2933-46b0-bf31-660cfa5621bd",
    "info": {},
    "user_keypair_id": "test",
    "management_public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCFe9ARO
+t9CybtuC1+cusDTeQL7wos1+U2dKPLCUJvNUN0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rj0wAQANhL7KatGLyCPQg9fq0oaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSLBrbTZkT0vEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9Ph0SqrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpdutFLH4eN5MFDh Generated-by-Sahara\n",
    "id": "e172d86c-906d-418e-a29c-6189f53bfa42",
    "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
    "node_groups": [
      {
        "image_id": null,
        "shares": null,
        "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
        "node_configs": {
          "YARN": {
            "yarn.nodemanager.vmem-check-enabled": "false",
            "yarn.scheduler.maximum-allocation-mb": 2048,
            "yarn.scheduler.minimum-allocation-mb": 256,
            "yarn.nodemanager.resource.memory-mb": 2048
          }
        }
      }
    ]
  }
}
```

```

        "MapReduce": {
            "yarn.app.mapreduce.am.resource.mb": 256,
            "mapreduce.task.io.sort.mb": 102,
            "mapreduce.reduce.java.opts": "-Xmx409m",
            "mapreduce.reduce.memory.mb": 512,
            "mapreduce.map.memory.mb": 256,
            "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
            "mapreduce.map.java.opts": "-Xmx204m"
        }
    },
    "auto_security_group": false,
    "availability_zone": null,
    "count": 1,
    "flavor_id": "2",
    "id": "0fe07f2a-0275-4bc0-93b2-c3c1e48e2815",
    "security_groups": null,
    "use_autoconfig": true,
    "instances": [],
    "volumes_availability_zone": null,
    "created_at": "2015-09-14T10:57:11",
    "node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
    "updated_at": "2015-09-14T10:57:12",
    "volumes_per_node": 0,
    "is_proxy_gateway": false,
    "name": "master",
    "volume_mount_prefix": "/volumes/disk",
    "node_processes": [
        "namenode",
        "resourcemanager",
        "oozie",
        "historyserver"
    ],
    "volumes_size": 0,
    "volume_local_to_instance": false,
    "volume_type": null
},
{
    "image_id": null,
    "shares": null,
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
    "node_configs": {
        "YARN": {
            "yarn.nodemanager.vmem-check-enabled": "false",
            "yarn.scheduler.maximum-allocation-mb": 2048,
            "yarn.scheduler.minimum-allocation-mb": 256,
            "yarn.nodemanager.resource.memory-mb": 2048
        },
        "MapReduce": {
            "yarn.app.mapreduce.am.resource.mb": 256,
            "mapreduce.task.io.sort.mb": 102,
            "mapreduce.reduce.java.opts": "-Xmx409m",
            "mapreduce.reduce.memory.mb": 512,
            "mapreduce.map.memory.mb": 256,
            "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
            "mapreduce.map.java.opts": "-Xmx204m"
        }
    },
    "auto_security_group": false,
    "availability_zone": null,

```

```

        "count": 3,
        "flavor_id": "2",
        "id": "c7a3bea4-c898-446b-8c67-6d378d4c06c4",
        "security_groups": null,
        "use_autoconfig": true,
        "instances": [],
        "volumes_availability_zone": null,
        "created_at": "2015-09-14T10:57:11",
        "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
        "updated_at": "2015-09-14T10:57:12",
        "volumes_per_node": 0,
        "is_proxy_gateway": false,
        "name": "worker",
        "volume_mount_prefix": "/volumes/disk",
        "node_processes": [
            "datanode",
            "nodemanager"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "volume_type": null
    },
    "provision_progress": [],
    "hadoop_version": "2.7.1",
    "use_autoconfig": true,
    "trust_id": null,
    "description": null,
    "created_at": "2015-09-14T10:57:11",
    "is_protected": false,
    "updated_at": "2015-09-14T10:57:12",
    "is_transient": false,
    "cluster_configs": {
        "HDFS": {
            "dfs.replication": 3
        }
    },
    "anti_affinity": [],
    "name": "vanilla-cluster",
    "default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
    "status": "Validating"
}

```

6.5.3. Create multiple clusters

Method	URI	Description
POST	/v1.1/{tenant_id}/clusters/multiple	Creates multiple clusters.

Normal response codes: 202

6.5.3.1. Request

Example 6.27. Create multiple clusters: JSON request

```
{
  "plugin_name": "vanilla",
  "hadoop_version": "2.6.0",
  "cluster_template_id": "9951f86d-57ba-43d6-9cb0-14ed2ec7a6cf",
  "default_image_id": "bc3c3d3c-2684-4bf8-a9fa-388fb71288a9",
  "user_keypair_id": "test",
  "name": "def-cluster",
  "count": 2,
  "cluster_configs": {},
  "neutron_management_network": "7e31648b-4b2e-4f32-9b0a-113581c27076"
}
```

6.5.3.2. Response

Example 6.28. Create multiple clusters: JSON response

```
{
  "clusters": [
    "a007a3e7-658f-4568-b0f2-fe2fd5efc554",
    "b012a6et-65hf-4566-b0f2-fe3fd7efc567"
  ]
}
```

6.5.4. Show details of a cluster

Method	URI	Description
GET	/v1.1/{tenant_id}/clusters/{cluster_id}	Shows details for a cluster, by ID.

Normal response codes: 200

6.5.4.1. Request

This table shows the URI parameters for the show details of a cluster request:

Name	Type	Description
{cluster_id}	UUID	The ID of the cluster

This operation does not accept a request body.

6.5.4.2. Response

Example 6.29. Show details of a cluster: JSON response

```
{
  "cluster": {
    "is_public": false,
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "shares": null,
    "status_description": "",
    "plugin_name": "vanilla",
    "neutron_management_network": "b1610452-2933-46b0-bf31-660cfa5621bd",
    "info": {},
    "user_keypair_id": "test",
    "management_public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACfe9AR0
+t9CybtuC1+cusDTeQL7wos1+U2dKPLCUJvNUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamck8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rj0wAQANhL7KatGLyCPQg9fq0oaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZkT0vEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9Ph0SqcrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpdutFLH4eN5MFDh Generated-by-Sahara\n",
    "id": "e172d86c-906d-418e-a29c-6189f53bfa42",
    "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
    "node_groups": [
      {
        "image_id": null,
        "shares": null,
        "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
        "node_configs": {
          "YARN": {
            "yarn.nodemanager.vmem-check-enabled": "false",
            "yarn.scheduler.maximum-allocation-mb": 2048,
            "yarn.scheduler.minimum-allocation-mb": 256,
            "yarn.nodemanager.resource.memory-mb": 2048
          },
          "MapReduce": {
            "yarn.app.mapreduce.am.resource.mb": 256,
            "mapreduce.task.io.sort.mb": 102,
            "mapreduce.reduce.java.opts": "-Xmx409m",
```



```

        "mapreduce.reduce.memory.mb": 512,
        "mapreduce.map.memory.mb": 256,
        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
        "mapreduce.map.java.opts": "-Xmx204m"
    }
},
"auto_security_group": false,
"availability_zone": null,
"count": 1,
"flavor_id": "2",
"id": "0fe07f2a-0275-4bc0-93b2-c3c1e48e2815",
"security_groups": null,
"use_autoconfig": true,
"instances": [],
"volumes_availability_zone": null,
"created_at": "2015-09-14T10:57:11",
"node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
"updated_at": "2015-09-14T10:57:12",
"volumes_per_node": 0,
"is_proxy_gateway": false,
"name": "master",
"volume_mount_prefix": "/volumes/disk",
"node_processes": [
    "namenode",
    "resourcemanager",
    "oozie",
    "historyserver"
],
"volumes_size": 0,
"volume_local_to_instance": false,
"volume_type": null
},
{
    "image_id": null,
    "shares": null,
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
    "node_configs": {
        "YARN": {
            "yarn.nodemanager.vmem-check-enabled": "false",
            "yarn.scheduler.maximum-allocation-mb": 2048,
            "yarn.scheduler.minimum-allocation-mb": 256,
            "yarn.nodemanager.resource.memory-mb": 2048
        },
        "MapReduce": {
            "yarn.app.mapreduce.am.resource.mb": 256,
            "mapreduce.task.io.sort.mb": 102,
            "mapreduce.reduce.java.opts": "-Xmx409m",
            "mapreduce.reduce.memory.mb": 512,
            "mapreduce.map.memory.mb": 256,
            "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
            "mapreduce.map.java.opts": "-Xmx204m"
        }
    },
    "auto_security_group": false,
    "availability_zone": null,
    "count": 3,
    "flavor_id": "2",
    "id": "c7a3bea4-c898-446b-8c67-6d378d4c06c4",
    "security_groups": null,

```

```

        "use_autoconfig": true,
        "instances": [],
        "volumes_availability_zone": null,
        "created_at": "2015-09-14T10:57:11",
        "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
        "updated_at": "2015-09-14T10:57:12",
        "volumes_per_node": 0,
        "is_proxy_gateway": false,
        "name": "worker",
        "volume_mount_prefix": "/volumes/disk",
        "node_processes": [
            "datanode",
            "nodemanager"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "volume_type": null
    },
    ],
    "provision_progress": [],
    "hadoop_version": "2.7.1",
    "use_autoconfig": true,
    "trust_id": null,
    "description": null,
    "created_at": "2015-09-14T10:57:11",
    "is_protected": false,
    "updated_at": "2015-09-14T10:57:12",
    "is_transient": false,
    "cluster_configs": {
        "HDFS": {
            "dfs.replication": 3
        }
    },
    "anti_affinity": [],
    "name": "vanilla-cluster",
    "default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
    "status": "Validating"
}

```

6.5.5. Scale cluster

Method	URI	Description
PUT	/v1.1/{tenant_id}/clusters/{cluster_id}	Scales a cluster.

Normal response codes: 202

6.5.5.1. Request

This table shows the URI parameters for the scale cluster request:

Name	Type	Description
{cluster_id}	UUID	The ID of the cluster

Example 6.30. Scale cluster: JSON request

```
{
  "add_node_groups": [
    {
      "count": 1,
      "name": "b-worker",
      "node_group_template_id": "bc270ffe-a086-4eeb-9baa-2f5a73504622"
    }
  ],
  "resize_node_groups": [
    {
      "count": 4,
      "name": "worker"
    }
  ]
}
```

6.5.5.2. Response

Example 6.31. Scale cluster: JSON response

```
{
  "cluster": {
    "info": {
      "YARN": {
        "Web UI": "http://172.18.168.115:8088",
        "ResourceManager": "http://172.18.168.115:8032"
      },
      "HDFS": {
        "Web UI": "http://172.18.168.115:50070",
        "NameNode": "hdfs://vanilla-cluster-master-0:9000"
      },
      "MapReduce JobHistory Server": {
        "Web UI": "http://172.18.168.115:19888"
      },
      "JobFlow": {
        "Oozie": "http://172.18.168.115:11000"
      }
    },
    "plugin_name": "vanilla",
    "hadoop_version": "2.7.1",
  }
}
```

```

    "updated_at": "2015-09-14T11:01:15",
    "name": "vanilla-cluster",
    "id": "e172d86c-906d-418e-a29c-6189f53bfa42",
    "management_public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACfe9AR0
+t9CybtuC1+cusDTeQL7wos1+U2dKPLCUJvNUN0PcunGefqWI4MUZPY9yGmvRqfINY7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rj0wAQANhL7KatGLyCPQg9fq0oaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKvk/
DPSlBrbTZkTOvEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9Ph0SqcrBbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpdutFLH4eN5MFDh Generated-by-Sahara\n",
    "trust_id": null,
    "status_description": "",
    "default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
    "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
    "is_protected": false,
    "is_transient": false,
    "provision_progress": [
      {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 1,
        "successful": true,
        "step_name": "Create Heat stack",
        "step_type": "Engine: create cluster",
        "updated_at": "2015-09-14T10:57:38",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T10:57:18",
        "id": "0a6d95f9-30f4-4434-823a-a38a7999a5af"
      },
      {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 4,
        "successful": true,
        "step_name": "Configure instances",
        "step_type": "Engine: create cluster",
        "updated_at": "2015-09-14T10:58:22",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T10:58:16",
        "id": "29f2b587-c34c-4871-9ed9-9235b411cd9a"
      },
      {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 1,
        "successful": true,
        "step_name": "Start the following process(es): Oozie",
        "step_type": "Plugin: start cluster",
        "updated_at": "2015-09-14T11:01:15",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T11:00:27",
        "id": "36f1efde-90f9-41c1-b409-aa1cf9623e3e"
      },
      {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 4,
        "successful": true,
        "step_name": "Configure instances",
        "step_type": "Plugin: configure cluster",
        "updated_at": "2015-09-14T10:59:21",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T10:58:22",

```

```

        "id": "602bcc27-3a2d-42c8-8aca-ebc475319c72"
      },
      {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 1,
        "successful": true,
        "step_name": "Configure topology data",
        "step_type": "Plugin: configure cluster",
        "updated_at": "2015-09-14T10:59:37",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T10:59:21",
        "id": "7e291df1-2d32-410d-ae89-33ab6f83cf17"
      },
      {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 3,
        "successful": true,
        "step_name": "Start the following process(es): DataNodes,
NodeManagers",
        "step_type": "Plugin: start cluster",
        "updated_at": "2015-09-14T11:00:11",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T11:00:01",
        "id": "8ab7933c-ad61-4a4f-88db-23ce78ee10f6"
      },
      {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 1,
        "successful": true,
        "step_name": "Await DataNodes start up",
        "step_type": "Plugin: start cluster",
        "updated_at": "2015-09-14T11:00:21",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T11:00:11",
        "id": "9c8dc016-8c5b-4e80-9857-80c41f6bd971"
      },
      {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 1,
        "successful": true,
        "step_name": "Start the following process(es): HistoryServer",
        "step_type": "Plugin: start cluster",
        "updated_at": "2015-09-14T11:00:27",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T11:00:21",
        "id": "c6327532-222b-416c-858f-73dbb32b8e97"
      },
      {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 4,
        "successful": true,
        "step_name": "Wait for instance accessibility",
        "step_type": "Engine: create cluster",
        "updated_at": "2015-09-14T10:58:14",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T10:57:41",
        "id": "d3eca726-8b44-473a-ac29-fba45a893725"
      },
      {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",

```

```

        "total": 0,
        "successful": true,
        "step_name": "Mount volumes to instances",
        "step_type": "Engine: create cluster",
        "updated_at": "2015-09-14T10:58:15",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T10:58:14",
        "id": "d7a875ff-64bf-41aa-882d-b5061c8ee152"
    },
    {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 1,
        "successful": true,
        "step_name": "Start the following process(es):
ResourceManager",
        "step_type": "Plugin: start cluster",
        "updated_at": "2015-09-14T11:00:00",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T10:59:55",
        "id": "ded7d227-10b8-4cb0-ab6c-25da1462bb7a"
    },
    {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 1,
        "successful": true,
        "step_name": "Start the following process(es): NameNode",
        "step_type": "Plugin: start cluster",
        "updated_at": "2015-09-14T10:59:54",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T10:59:38",
        "id": "e1701ff5-930a-4212-945a-43515dfe24d1"
    },
    {
        "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "total": 4,
        "successful": true,
        "step_name": "Assign IPs",
        "step_type": "Engine: create cluster",
        "updated_at": "2015-09-14T10:57:41",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "created_at": "2015-09-14T10:57:38",
        "id": "eaf0ab1b-bf8f-48f0-8f2c-fa4f82f539b9"
    }
],
"status": "Active",
"description": null,
"use_autoconfig": true,
"shares": null,
"neutron_management_network": "b1610452-2933-46b0-bf31-660cfa5621bd",
"is_public": false,
"tenant_id": "808d5032ea0446889097723bfc8e919d",
"node_groups": [
    {
        "volumes_per_node": 0,
        "volume_type": null,
        "updated_at": "2015-09-14T10:57:37",
        "name": "b-worker",
        "id": "b7a6dea4-c898-446b-8c67-4f378d4c06c4",
        "node_group_template_id": "bc270ffe-
a086-4eeb-9baa-2f5a73504622",

```

```

    "node_configs": {
      "YARN": {
        "yarn.nodemanager.vmem-check-enabled": "false",
        "yarn.scheduler.minimum-allocation-mb": 256,
        "yarn.nodemanager.resource.memory-mb": 2048,
        "yarn.scheduler.maximum-allocation-mb": 2048
      },
      "MapReduce": {
        "mapreduce.map.memory.mb": 256,
        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
        "mapreduce.map.java.opts": "-Xmx204m",
        "mapreduce.reduce.memory.mb": 512,
        "mapreduce.task.io.sort.mb": 102,
        "mapreduce.reduce.java.opts": "-Xmx409m",
        "yarn.app.mapreduce.am.resource.mb": 256
      }
    },
    "auto_security_group": false,
    "volumes_availability_zone": null,
    "use_autoconfig": true,
    "security_groups": null,
    "shares": null,
    "node_processes": [
      "datanode",
      "nodemanager"
    ],
    "availability_zone": null,
    "flavor_id": "2",
    "image_id": null,
    "volume_local_to_instance": false,
    "count": 1,
    "volumes_size": 0,
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
    "volume_mount_prefix": "/volumes/disk",
    "instances": [],
    "is_proxy_gateway": false,
    "created_at": "2015-09-14T10:57:11"
  },
  {
    "volumes_per_node": 0,
    "volume_type": null,
    "updated_at": "2015-09-14T10:57:36",
    "name": "master",
    "id": "0fe07f2a-0275-4bc0-93b2-c3c1e48e2815",
    "node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
    "node_configs": {
      "YARN": {
        "yarn.nodemanager.vmem-check-enabled": "false",
        "yarn.scheduler.minimum-allocation-mb": 256,
        "yarn.nodemanager.resource.memory-mb": 2048,
        "yarn.scheduler.maximum-allocation-mb": 2048
      },
      "MapReduce": {
        "mapreduce.map.memory.mb": 256,
        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
        "mapreduce.map.java.opts": "-Xmx204m",
        "mapreduce.reduce.memory.mb": 512,
        "mapreduce.task.io.sort.mb": 102,
        "mapreduce.reduce.java.opts": "-Xmx409m",

```

```

        "yarn.app.mapreduce.am.resource.mb": 256
      }
    },
    "auto_security_group": false,
    "volumes_availability_zone": null,
    "use_autoconfig": true,
    "security_groups": null,
    "shares": null,
    "node_processes": [
      "namenode",
      "resourcemanager",
      "oozie",
      "historyserver"
    ],
    "availability_zone": null,
    "flavor_id": "2",
    "image_id": null,
    "volume_local_to_instance": false,
    "count": 1,
    "volumes_size": 0,
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
    "volume_mount_prefix": "/volumes/disk",
    "instances": [
      {
        "instance_id": "b9f16a07-88fc-423e-83a3-489598fe6737",
        "internal_ip": "10.50.0.60",
        "instance_name": "vanilla-cluster-master-0",
        "updated_at": "2015-09-14T10:57:39",
        "management_ip": "172.18.168.115",
        "created_at": "2015-09-14T10:57:36",
        "id": "4867d92e-cc7b-4cde-9a1a-149e91caa491"
      }
    ],
    "is_proxy_gateway": false,
    "created_at": "2015-09-14T10:57:11"
  },
  {
    "volumes_per_node": 0,
    "volume_type": null,
    "updated_at": "2015-09-14T10:57:37",
    "name": "worker",
    "id": "c7a3bea4-c898-446b-8c67-6d378d4c06c4",
    "node_group_template_id": "846edb31-add5-46e6-a4ee-a4c339f99251",
    "node_configs": {
      "YARN": {
        "yarn.nodemanager.vmem-check-enabled": "false",
        "yarn.scheduler.minimum-allocation-mb": 256,
        "yarn.nodemanager.resource.memory-mb": 2048,
        "yarn.scheduler.maximum-allocation-mb": 2048
      },
      "MapReduce": {
        "mapreduce.map.memory.mb": 256,
        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
        "mapreduce.map.java.opts": "-Xmx204m",
        "mapreduce.reduce.memory.mb": 512,
        "mapreduce.task.io.sort.mb": 102,
        "mapreduce.reduce.java.opts": "-Xmx409m",
        "yarn.app.mapreduce.am.resource.mb": 256
      }
    }
  }
}

```

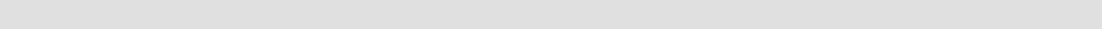


```

    },
    "auto_security_group": false,
    "volumes_availability_zone": null,
    "use_autoconfig": true,
    "security_groups": null,
    "shares": null,
    "node_processes": [
      "datanode",
      "nodemanager"
    ],
    "availability_zone": null,
    "flavor_id": "2",
    "image_id": null,
    "volume_local_to_instance": false,
    "count": 4,
    "volumes_size": 0,
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
    "volume_mount_prefix": "/volumes/disk",
    "instances": [
      {
        "instance_id": "0cf1ee81-aa72-48da-be2c-65bc2fa51f8f",
        "internal_ip": "10.50.0.63",
        "instance_name": "vanilla-cluster-worker-0",
        "updated_at": "2015-09-14T10:57:39",
        "management_ip": "172.18.168.118",
        "created_at": "2015-09-14T10:57:37",
        "id": "f3633b30-c1e4-4144-930b-ab5b780b87be"
      },
      {
        "instance_id": "4a937391-b594-4ad0-9a53-00a99a691383",
        "internal_ip": "10.50.0.62",
        "instance_name": "vanilla-cluster-worker-1",
        "updated_at": "2015-09-14T10:57:40",
        "management_ip": "172.18.168.117",
        "created_at": "2015-09-14T10:57:37",
        "id": "0d66fd93-f277-4a94-b46a-f5866aa0c38f"
      },
      {
        "instance_id": "839b1d56-6d0d-4aa4-9d05-30e029c276f8",
        "internal_ip": "10.50.0.61",
        "instance_name": "vanilla-cluster-worker-2",
        "updated_at": "2015-09-14T10:57:40",
        "management_ip": "172.18.168.116",
        "created_at": "2015-09-14T10:57:37",
        "id": "0982cefd-5c58-436e-8f1e-c1d0830f18a7"
      }
    ],
    "is_proxy_gateway": false,
    "created_at": "2015-09-14T10:57:11"
  },
  "cluster_configs": {
    "HDFS": {
      "dfs.replication": 3
    }
  },
  "user_keypair_id": "apavlov",
  "anti_affinity": [],
  "created_at": "2015-09-14T10:57:11"
}

```

```
}
```



6.5.6. Update cluster

Method	URI	Description
PATCH	/v1.1/{tenant_id}/clusters/{cluster_id}	Updates a cluster.

Normal response codes: 202

6.5.6.1. Request

This table shows the URI parameters for the update cluster request:

Name	Type	Description
{cluster_id}	UUID	The ID of the cluster

Example 6.32. Update cluster: JSON request

```
{
  "name": "public-vanilla-cluster",
  "is_public": true
}
```

6.5.6.2. Response

Example 6.33. Update cluster: JSON response

```
{
  "cluster": {
    "is_public": true,
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "shares": null,
    "status_description": "",
    "plugin_name": "vanilla",
    "neutron_management_network": "b1610452-2933-46b0-bf31-660cfa5621bd",
    "info": {},
    "user_keypair_id": "test",
    "management_public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCFe9AR0
+t9CybtuC1+cusDTeQL7wos1+U2dKPLCUJvNUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rj0wAQANhL7KatGLyCPQg9fq0oaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKvk/
DPSlBrbTZkT0vEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9Ph0SqrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpdutFLH4eN5MFDh Generated-by-Sahara\n",
    "id": "e172d86c-906d-418e-a29c-6189f53bfa42",
    "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
    "node_groups": [
      {
        "image_id": null,
        "shares": null,
        "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
        "node_configs": {
          "YARN": {
            "yarn.nodemanager.vmem-check-enabled": "false",
            "yarn.scheduler.maximum-allocation-mb": 2048,
            "yarn.scheduler.minimum-allocation-mb": 256,
            "yarn.nodemanager.resource.memory-mb": 2048
          }
        }
      }
    ]
  }
}
```

```

    },
    "MapReduce": {
      "yarn.app.mapreduce.am.resource.mb": 256,
      "mapreduce.task.io.sort.mb": 102,
      "mapreduce.reduce.java.opts": "-Xmx409m",
      "mapreduce.reduce.memory.mb": 512,
      "mapreduce.map.memory.mb": 256,
      "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
      "mapreduce.map.java.opts": "-Xmx204m"
    }
  },
  "auto_security_group": false,
  "availability_zone": null,
  "count": 1,
  "flavor_id": "2",
  "id": "0fe07f2a-0275-4bc0-93b2-c3c1e48e2815",
  "security_groups": null,
  "use_autoconfig": true,
  "instances": [],
  "volumes_availability_zone": null,
  "created_at": "2015-09-14T10:57:11",
  "node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
  "updated_at": "2015-09-14T10:57:12",
  "volumes_per_node": 0,
  "is_proxy_gateway": false,
  "name": "master",
  "volume_mount_prefix": "/volumes/disk",
  "node_processes": [
    "namenode",
    "resourcemanager",
    "oozie",
    "historyserver"
  ],
  "volumes_size": 0,
  "volume_local_to_instance": false,
  "volume_type": null
},
{
  "image_id": null,
  "shares": null,
  "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
  "node_configs": {
    "YARN": {
      "yarn.nodemanager.vmem-check-enabled": "false",
      "yarn.scheduler.maximum-allocation-mb": 2048,
      "yarn.scheduler.minimum-allocation-mb": 256,
      "yarn.nodemanager.resource.memory-mb": 2048
    },
    "MapReduce": {
      "yarn.app.mapreduce.am.resource.mb": 256,
      "mapreduce.task.io.sort.mb": 102,
      "mapreduce.reduce.java.opts": "-Xmx409m",
      "mapreduce.reduce.memory.mb": 512,
      "mapreduce.map.memory.mb": 256,
      "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
      "mapreduce.map.java.opts": "-Xmx204m"
    }
  },
  "auto_security_group": false,

```

```

        "availability_zone": null,
        "count": 3,
        "flavor_id": "2",
        "id": "c7a3bea4-c898-446b-8c67-6d378d4c06c4",
        "security_groups": null,
        "use_autoconfig": true,
        "instances": [],
        "volumes_availability_zone": null,
        "created_at": "2015-09-14T10:57:11",
        "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
        "updated_at": "2015-09-14T10:57:12",
        "volumes_per_node": 0,
        "is_proxy_gateway": false,
        "name": "worker",
        "volume_mount_prefix": "/volumes/disk",
        "node_processes": [
            "datanode",
            "nodemanager"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "volume_type": null
    },
    "provision_progress": [],
    "hadoop_version": "2.7.1",
    "use_autoconfig": true,
    "trust_id": null,
    "description": null,
    "created_at": "2015-09-14T10:57:11",
    "is_protected": false,
    "updated_at": "2015-09-14T10:57:12",
    "is_transient": false,
    "cluster_configs": {
        "HDFS": {
            "dfs.replication": 3
        }
    },
    "anti_affinity": [],
    "name": "public-vanilla-cluster",
    "default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
    "status": "Validating"
}

```

6.5.7. Show progress

Method	URI	Description
GET	/v1.1/{tenant_id}/clusters/{cluster_id}	Shows provisioning progress for a cluster.

Normal response codes: 200

6.5.7.1. Request

This table shows the URI parameters for the show progress request:

Name	Type	Description
{cluster_id}	UUID	The ID of the cluster

6.5.7.2. Response

Example 6.34. Show progress: JSON response

```
{
  "status": "Error",
  "neutron_management_network": "7e31648b-4b2e-4f32-9b0a-113581c27076",
  "is_transient": false,
  "description": "",
  "user_keypair_id": "vgridnev",
  "updated_at": "2015-03-31 14:10:59",
  "plugin_name": "spark",
  "provision_progress": [
    {
      "successful": false,
      "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
      "created_at": "2015-03-31 14:10:20",
      "step_type": "Engine: create cluster",
      "updated_at": "2015-03-31 14:10:35",
      "events": [
        {
          "instance_name": "sample-worker-spark-004",
          "successful": false,
          "created_at": "2015-03-31 14:10:35",
          "updated_at": null,
          "event_info": "Node sample-worker-spark-004 has error
status\nError ID: 3e238c82-d1f5-4560-8ed8-691e923e16a0",
          "instance_id": "b5ba5ba8-e9c1-47f7-9355-3ce0ec0e449d",
          "node_group_id": "145cf2fb-dcdf-42af-a4b9-a4047d2919d4",
          "step_id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6",
          "id": "34afcfc7-bdb0-43cb-b142-283d560dc6ad"
        },
        {
          "instance_name": "sample-worker-spark-001",
          "successful": true,
          "created_at": "2015-03-31 14:10:35",
          "updated_at": null,
          "event_info": null,
          "instance_id": "c532ab71-38da-475a-95f8-f8eb93b8f1c2",
          "node_group_id": "145cf2fb-dcdf-42af-a4b9-a4047d2919d4",

```

```

        "step_id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6",
        "id": "4ba50414-5216-4161-bc7a-12716122b99d"
    },
    {
        "cluster_id": "c26ec982-ba6b-4d75-818c-a50240164af0",
        "step_name": "Wait for instances to become active",
        "total": 5,
        "id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6"
    },
    {
        "successful": true,
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "created_at": "2015-03-31 14:10:12",
        "step_type": "Engine: create cluster",
        "updated_at": "2015-03-31 14:10:19",
        "events": [],
        "cluster_id": "c26ec982-ba6b-4d75-818c-a50240164af0",
        "step_name": "Run instances",
        "total": 5,
        "id": "407ba50a-c799-46af-9dfb-6aa5f6ade426"
    }
],
"anti_affinity": [],
"node_groups": [],
"management_public_key": "Sahara",
"status_description": "Creating cluster failed for the following
reason(s): Node sample-worker-spark-004 has error status\nError ID: 3e238c82-
d1f5-4560-8ed8-691e923e16a0",
"hadoop_version": "1.0.0",
"id": "c26ec982-ba6b-4d75-1f8c-a50240164af0",
"trust_id": null,
"info": {},
"cluster_template_id": "5a9a09a3-9349-43bd-9058-16c401fad2d5",
"name": "sample",
"cluster_configs": {},
"created_at": "2015-03-31 14:10:07",
"default_image_id": "e6a6c5da-67be-4017-a7d2-81f466efe67e",
"tenant_id": "9cd1314a0a31493282b6712b76a8fcda"
}

```

6.5.8. Delete a cluster

Method	URI	Description
DELETE	/v1.1/{tenant_id}/clusters/{cluster_id}	Deletes a cluster.

Normal response codes: 204

6.5.8.1. Request

This table shows the URI parameters for the delete a cluster request:

Name	Type	Description
{cluster_id}	UUID	The ID of the cluster

This operation does not accept a request body.

6.6. Event log

The event log feature provides information about cluster provisioning. In the event of errors, the event log shows the reason for the failure.

Method	URI	Description
GET	/v1.1/{tenant_id}/clusters/{cluster_id}	Shows provisioning progress of cluster.

6.6.1. Show progress

Method	URI	Description
GET	/v1.1/{tenant_id}/clusters/{cluster_id}	Shows provisioning progress of cluster.

Normal response codes: 200

6.6.1.1. Request

This table shows the URI parameters for the show progress request:

Name	Type	Description
{cluster_id}	UUID	The UUID of the cluster

6.6.1.2. Response

Example 6.35. Show progress: JSON response

```
{
  "status": "Error",
  "neutron_management_network": "7e31648b-4b2e-4f32-9b0a-113581c27076",
  "is_transient": false,
  "description": "",
  "user_keypair_id": "vgridnev",
  "updated_at": "2015-03-31 14:10:59",
  "plugin_name": "spark",
  "provision_progress": [
    {
      "successful": false,
      "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
      "created_at": "2015-03-31 14:10:20",
      "step_type": "Engine: create cluster",
      "updated_at": "2015-03-31 14:10:35",
      "events": [
        {
          "instance_name": "sample-worker-spark-004",
          "successful": false,
          "created_at": "2015-03-31 14:10:35",
          "updated_at": null,
          "event_info": "Node sample-worker-spark-004 has error
status\nError ID: 3e238c82-d1f5-4560-8ed8-691e923e16a0",
          "instance_id": "b5ba5ba8-e9c1-47f7-9355-3ce0ec0e449d",
          "node_group_id": "145cf2fb-dcdf-42af-a4b9-a4047d2919d4",
          "step_id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6",
          "id": "34afcfc7-bdb0-43cb-b142-283d560dc6ad"
        },
        {
          "instance_name": "sample-worker-spark-001",
          "successful": true,
          "created_at": "2015-03-31 14:10:35",
          "updated_at": null,
          "event_info": null,
          "instance_id": "c532ab71-38da-475a-95f8-f8eb93b8f1c2",
          "node_group_id": "145cf2fb-dcdf-42af-a4b9-a4047d2919d4",
```

```

        "step_id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6",
        "id": "4ba50414-5216-4161-bc7a-12716122b99d"
    },
    {
        "cluster_id": "c26ec982-ba6b-4d75-818c-a50240164af0",
        "step_name": "Wait for instances to become active",
        "total": 5,
        "id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6"
    },
    {
        "successful": true,
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "created_at": "2015-03-31 14:10:12",
        "step_type": "Engine: create cluster",
        "updated_at": "2015-03-31 14:10:19",
        "events": [],
        "cluster_id": "c26ec982-ba6b-4d75-818c-a50240164af0",
        "step_name": "Run instances",
        "total": 5,
        "id": "407ba50a-c799-46af-9dfb-6aa5f6ade426"
    }
],
"anti_affinity": [],
"node_groups": [],
"management_public_key": "Sahara",
"status_description": "Creating cluster failed for the following
reason(s): Node sample-worker-spark-004 has error status\nError ID: 3e238c82-
d1f5-4560-8ed8-691e923e16a0",
"hadoop_version": "1.0.0",
"id": "c26ec982-ba6b-4d75-1f8c-a50240164af0",
"trust_id": null,
"info": {},
"cluster_template_id": "5a9a09a3-9349-43bd-9058-16c401fad2d5",
"name": "sample",
"cluster_configs": {},
"created_at": "2015-03-31 14:10:07",
"default_image_id": "e6a6c5da-67be-4017-a7d2-81f466efe67e",
"tenant_id": "9cd1314a0a31493282b6712b76a8fcda"
}

```

6.7. Data sources

A data source object defines the location of input or output for MapReduce jobs and might reference different types of storage.

The Data Processing service does not validate data source locations.

Method	URI	Description
GET	/v1.1/{tenant_id}/data-sources	Lists all data sources.
POST	/v1.1/{tenant_id}/data-sources	Creates a data source.
GET	/v1.1/{tenant_id}/data-sources/{data_source_id}	Shows details for a data source.
PUT	/v1.1/{tenant_id}/data-sources/{data_source_id}	Updates a data source.
DELETE	/v1.1/{tenant_id}/data-sources/{data_source_id}	Deletes a data source.

6.7.1. List data sources

Method	URI	Description
GET	/v1.1/{tenant_id}/data-sources	Lists all data sources.

Normal response codes: 200

6.7.1.1. Request

This operation does not accept a request body.

6.7.1.2. Response

Example 6.36. List data sources: JSON response

```
{
  "data_sources": [
    {
      "is_public": false,
      "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
      "is_protected": false,
      "created_at": "2015-03-26 11:18:10",
      "id": "953831f2-0852-49d8-ac71-af5805e25256",
      "name": "swift_input",
      "updated_at": null,
      "description": "This is input",
      "url": "swift://container/text",
      "type": "swift"
    },
    {
      "is_public": false,
      "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
      "is_protected": false,
      "created_at": "2015-03-26 11:09:36",
      "id": "d7fffe9c-3b42-46a9-8be8-e98f586fa7a9",
      "name": "hdfs_input",
      "updated_at": null,
      "description": "This is hdfs input",
      "url": "hdfs://test-master-node:8020/user/hadoop/input",
      "type": "hdfs"
    }
  ]
}
```

6.7.2. Create data source

Method	URI	Description
POST	/v1.1/{tenant_id}/data-sources	Creates a data source.

Normal response codes: 202

6.7.2.1. Request

Example 6.37. Create data source: JSON request

```
{
  "description": "This is input",
  "url": "swift://container/text",
  "credentials": {
    "password": "swordfish",
    "user": "dev"
  },
  "type": "swift",
  "name": "swift_input"
}
```

Example 6.38. Create data source: JSON request

```
{
  "description": "This is hdfs input",
  "url": "hdfs://test-master-node:8020/user/hadoop/input",
  "type": "hdfs",
  "name": "hdfs_input"
}
```

6.7.2.2. Response

Example 6.39. Create data source: JSON response

```
{
  "data_source": {
    "is_public": false,
    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
    "is_protected": false,
    "created_at": "2015-03-26 11:18:10.691493",
    "id": "953831f2-0852-49d8-ac71-af5805e25256",
    "updated_at": null,
    "name": "swift_input",
    "description": "This is input",
    "url": "swift://container/text",
    "type": "swift"
  }
}
```

Example 6.40. Create data source: JSON response

```
{
  "data_source": {
    "is_public": false,
```

```
    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",  
    "is_protected": false,  
    "created_at": "2015-03-26 11:09:36.148464",  
    "id": "d7fffe9c-3b42-46a9-8be8-e98f586fa7a9",  
    "updated_at": null,  
    "name": "hdfs_input",  
    "description": "This is hdfs input",  
    "url": "hdfs://test-master-node:8020/user/hadoop/input",  
    "type": "hdfs"  
  }  
}
```

6.7.3. Show data source details

Method	URI	Description
GET	/v1.1/{tenant_id}/data-sources/{data_source_id}	Shows details for a data source.

Normal response codes: 200

6.7.3.1. Request

This table shows the URI parameters for the show data source details request:

Name	Type	Description
{data_source_id}	UUID	The UUID of the data source.

This operation does not accept a request body.

6.7.3.2. Response

Example 6.41. Show data source details: JSON response

```
{
  "data_source": {
    "is_public": false,
    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
    "is_protected": false,
    "created_at": "2015-03-26 11:18:10.691493",
    "id": "953831f2-0852-49d8-ac71-af5805e25256",
    "updated_at": null,
    "name": "swift_input",
    "description": "This is input",
    "url": "swift://container/text",
    "type": "swift"
  }
}
```

6.7.4. Update data source

Method	URI	Description
PUT	/v1.1/{tenant_id}/data-sources/{data_source_id}	Updates a data source.

Normal response codes: 202

6.7.4.1. Request

This table shows the URI parameters for the update data source request:

Name	Type	Description
{data_source_id}	UUID	The UUID of the data source.

Example 6.42. Update data source: JSON request

```
{
  "description": "This is public input",
  "is_protected": true
}
```

6.7.4.2. Response

Example 6.43. Update data source: JSON response

```
{
  "data_source": {
    "is_public": true,
    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
    "is_protected": false,
    "created_at": "2015-09-15 12:32:24.847493",
    "id": "953831f2-0852-49d8-ac71-af5805e25256",
    "updated_at": "2015-09-15 12:34:42.597435",
    "name": "swift_input",
    "description": "This is public input",
    "url": "swift://container/text",
    "type": "swift"
  }
}
```

6.7.5. Delete data source

Method	URI	Description
DELETE	/v1.1/{tenant_id}/data-sources/{data_source_id}	Deletes a data source.

Normal response codes: 204

6.7.5.1. Request

This table shows the URI parameters for the delete data source request:

Name	Type	Description
{data_source_id}	UUID	The UUID of the data source.

This operation does not accept a request body.

6.8. Job binary internals

Job binary internal objects represent data processing applications and libraries that are stored in the internal database.

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binary-internals	Lists the available job binary internals.
PUT	/v1.1/{tenant_id}/job-binary-internals/{name}	Creates a job binary internal.
GET	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}	Shows details for a job binary internal.
DELETE	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}	Deletes a job binary internal.
PATCH	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}	Updates a job binary internal.
GET	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}/data	Shows data for a job binary internal.

6.8.1. List job binary internals

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binary-internals	Lists the available job binary internals.

Normal response codes: 200

6.8.1.1. Request

This operation does not accept a request body.

6.8.1.2. Response

Example 6.44. List job binary internals: JSON response

```
{
  "binaries": [
    {
      "is_public": false,
      "name": "example.pig",
      "tenant_id": "11587919cc534bcbb1027a161c82cf58",
      "created_at": "2013-10-15 12:36:59.329034",
      "updated_at": null,
      "datasize": 161,
      "id": "d2498cbf-4589-484a-a814-81436c18beb3",
      "is_protected": false
    },
    {
      "is_public": false,
      "name": "udf.jar",
      "tenant_id": "11587919cc534bcbb1027a161c82cf58",
      "created_at": "2013-10-15 12:43:52.008620",
      "updated_at": null,
      "datasize": 3745,
      "id": "22f1d87a-23c8-483e-a0dd-cb4a16dde5f9",
      "is_protected": false
    }
  ]
}
```

6.8.2. Create job binary internal

Method	URI	Description
PUT	/v1.1/{tenant_id}/job-binary-internals/{name}	Creates a job binary internal.

Job binary internals are objects that represent data processing applications and libraries that are stored in the internal database.

Specify the file contents (raw data or script text) in the request body. Specify the file name in the URI.

Normal response codes: 202

6.8.2.1. Request

This table shows the URI parameters for the create job binary internal request:

Name	Type	Description
{name}	String	The name of the job binary internal.

This operation does not accept a request body.

6.8.2.2. Response

Example 6.45. Create job binary internal: JSON response

```
{
  "job_binary_internal": {
    "is_public": false,
    "name": "script.pig",
    "tenant_id": "11587919cc534bcbb1027a161c82cf58",
    "created_at": "2013-10-15 13:17:35.994466",
    "updated_at": null,
    "datasize": 160,
    "id": "4833dc4b-8682-4d5b-8a9f-2036b47a0996",
    "is_protected": false
  }
}
```

6.8.3. Show job binary internal details

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}	Shows details for a job binary internal.

Normal response codes: 200

6.8.3.1. Request

This table shows the URI parameters for the show job binary internal details request:

Name	Type	Description
{job_binary_internals_id}	UUID	The UUID of the job binary internal.

This operation does not accept a request body.

6.8.3.2. Response

Example 6.46. Show job binary internal details: JSON response

```
{
  "job_binary_internal": {
    "is_public": false,
    "name": "script.pig",
    "tenant_id": "11587919cc534bcbb1027a161c82cf58",
    "created_at": "2013-10-15 13:17:35.994466",
    "updated_at": null,
    "datasize": 160,
    "id": "4833dc4b-8682-4d5b-8a9f-2036b47a0996",
    "is_protected": false
  }
}
```

6.8.4. Delete job binary internal

Method	URI	Description
DELETE	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}	Deletes a job binary internal.

Normal response codes: 204

6.8.4.1. Request

This table shows the URI parameters for the delete job binary internal request:

Name	Type	Description
{job_binary_internals_id}	UUID	The UUID of the job binary internal.

This operation does not accept a request body.

6.8.5. Update job binary internal

Method	URI	Description
PATCH	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}	Updates a job binary internal.

Normal response codes: 202

6.8.5.1. Request

This table shows the URI parameters for the update job binary internal request:

Name	Type	Description
{job_binary_internals_id}	UUID	The UUID of the job binary internal.

Example 6.47. Update job binary internal : JSON request

```
{
  "name": "public-jbi",
  "is_public": true
}
```

6.8.5.2. Response

Example 6.48. Update job binary internal: JSON response

```
{
  "job_binary_internal": {
    "is_public": true,
    "name": "public-jbi",
    "tenant_id": "11587919cc534bcbb1027a161c82cf58",
    "created_at": "2015-09-15 13:21:54.485912",
    "updated_at": "2015-09-15 13:24:24.590124",
    "datasize": 200,
    "id": "2433dc4b-8682-4d5b-8a9f-2036d47a0996",
    "is_protected": false
  }
}
```

6.8.6. Show job binary internal data

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}/data	Shows data for a job binary internal.

The response body shows the job binary raw data and the response headers show the data length.

Example response:

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 161
Content-Type: text/html; charset=utf-8
Date: Sat, 28 Mar 2016 02:21:13 GMT

A = load '$INPUT' using PigStorage(':') as (fruit: chararray);
B = foreach A generate com.hadoopbook.pig.Trim(fruit);
store B into '$OUTPUT' USING PigStorage();
```

Normal response codes: 200

6.8.6.1. Request

This table shows the URI parameters for the show job binary internal data request:

Name	Type	Description
{job_binary_internals_id}	UUID	The UUID of the job binary internal.

This operation does not accept a request body.

6.8.6.2. Response

This operation does not return a response body.

6.9. Job binaries

Job binary objects represent data processing applications and libraries that are stored in either the internal database or the Object Storage service.

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binaries	Lists the available job binaries.
GET	/v1.1/{tenant_id}/job-binaries	Shows details for a job binary.
POST	/v1.1/{tenant_id}/job-binaries	Creates a job binary.
PUT	/v1.1/{tenant_id}/job-binaries	Updates a job binary.
DELETE	/v1.1/{tenant_id}/job-binaries	Deletes a job binary.
GET	/v1.1/{tenant_id}/job-binaries/{job_binary_id}/data	Shows data for a job binary.

6.9.1. List job binaries

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binaries	Lists the available job binaries.

Normal response codes: 200

6.9.1.1. Request

This operation does not accept a request body.

6.9.1.2. Response

Example 6.49. List job binaries: JSON response

```
{
  "binaries": [
    {
      "is_public": false,
      "description": "",
      "url": "internal-db://d2498cbf-4589-484a-a814-81436c18beb3",
      "tenant_id": "11587919cc534bcbb1027a161c82cf58",
      "created_at": "2013-10-15 12:36:59.375060",
      "updated_at": null,
      "id": "84248975-3c82-4206-a58d-6e7fb3a563fd",
      "name": "example.pig",
      "is_protected": false
    },
    {
      "is_public": false,
      "description": "",
      "url": "internal-db://22f1d87a-23c8-483e-a0dd-cb4a16dde5f9",
      "tenant_id": "11587919cc534bcbb1027a161c82cf58",
      "created_at": "2013-10-15 12:43:52.265899",
      "updated_at": null,
      "id": "508fc62d-1d58-4412-b603-bdab307bb926",
      "name": "udf.jar",
      "is_protected": false
    },
    {
      "is_public": false,
      "description": "",
      "url": "swift://container/jar-example.jar",
      "tenant_id": "11587919cc534bcbb1027a161c82cf58",
      "created_at": "2013-10-15 14:25:04.970513",
      "updated_at": null,
      "id": "a716a9cd-9add-4b12-b1b6-cdb71aaef350",
      "name": "jar-example.jar",
      "is_protected": false
    }
  ]
}
```

6.9.2. Show job binary details

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binaries	Shows details for a job binary.

Normal response codes: 200

6.9.2.1. Request

This operation does not accept a request body.

6.9.2.2. Response

Example 6.50. Show job binary details: JSON response

```
{
  "job_binary": {
    "is_public": false,
    "description": "an example jar file",
    "url": "swift://container/jar-example.jar",
    "tenant_id": "11587919cc534bcbb1027a161c82cf58",
    "created_at": "2013-10-15 14:25:04.970513",
    "updated_at": null,
    "id": "a716a9cd-9add-4b12-b1b6-cdb71aaef350",
    "name": "jar-example.jar",
    "is_protected": false
  }
}
```


6.9.3. Create job binary

Method	URI	Description
POST	/v1.1/{tenant_id}/job-binaries	Creates a job binary.

Normal response codes: 202

6.9.3.1. Request

Example 6.51. Create job binary: JSON request

```
{
  "url": "swift://container/jar-example.jar",
  "name": "jar-example.jar",
  "description": "This is a job binary",
  "extra": {
    "password": "swordfish",
    "user": "admin"
  }
}
```

6.9.3.2. Response

Example 6.52. Create job binary: JSON response

```
{
  "job_binary": {
    "is_public": false,
    "description": "This is a job binary",
    "url": "swift://container/jar-example.jar",
    "tenant_id": "11587919cc534bcbb1027a161c82cf58",
    "created_at": "2013-10-15 14:49:20.106452",
    "id": "07f86352-ee8a-4b08-b737-d705ded5ff9c",
    "updated_at": null,
    "name": "jar-example.jar",
    "is_protected": false
  }
}
```

6.9.4. Update job binary

Method	URI	Description
PUT	/v1.1/{tenant_id}/job-binaries	Updates a job binary.

Normal response codes: 202

6.9.4.1. Request

Example 6.53. Update job binary : JSON request

```
{
  "url": "swift://container/new-jar-example.jar",
  "name": "new-jar-example.jar",
  "description": "This is a new job binary"
}
```

6.9.4.2. Response

Example 6.54. Update job binary: JSON response

```
{
  "job_binary": {
    "is_public": false,
    "description": "This is a new job binary",
    "url": "swift://container/new-jar-example.jar",
    "tenant_id": "11587919cc534bcbb1027a161c82cf58",
    "created_at": "2015-09-15 12:42:51.421542",
    "updated_at": null,
    "id": "b713d7ad-4add-4f12-g1b6-cdg71aaef350",
    "name": "new-jar-example.jar",
    "is_protected": false
  }
}
```

6.9.5. Delete job binary

Method	URI	Description
DELETE	/v1.1/{tenant_id}/job-binaries	Deletes a job binary.

Normal response codes: 204

6.9.5.1. Request

This operation does not accept a request body.

6.9.6. Show job binary data

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binaries/{job_binary_id}/data	Shows data for a job binary.

The response body shows the job binary raw data and the response headers show the data length.

Example response:

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 161
Content-Type: text/html; charset=utf-8
Date: Sat, 28 Mar 2016 02:42:48 GMT

A = load '$INPUT' using PigStorage(':') as (fruit: chararray);
B = foreach A generate com.hadoopbook.pig.Trim(fruit);
store B into '$OUTPUT' USING PigStorage();
```

Normal response codes: 200

6.9.6.1. Request

This table shows the URI parameters for the show job binary data request:

Name	Type	Description
{job_binary_id}	UUID	The UUID of the job binary.

This operation does not accept a request body.

6.9.6.2. Response

This operation does not return a response body.

6.10. Jobs

A job object lists the binaries that a job needs to run. To run a job, you must specify data sources and job parameters.

You can run a job on an existing or new transient cluster.

Method	URI	Description
GET	/v1.1/{tenant_id}/jobs	Lists all jobs.
POST	/v1.1/{tenant_id}/jobs	Creates a job object.
GET	/v1.1/{tenant_id}/jobs/{job_id}	Shows details for a job.
DELETE	/v1.1/{tenant_id}/jobs/{job_id}	Removes a job.
PATCH	/v1.1/{tenant_id}/jobs/{job_id}	Updates a job object.
POST	/v1.1/{tenant_id}/jobs/{job_id}/execute	Runs a job.

6.10.1. List jobs

Method	URI	Description
GET	/v1.1/{tenant_id}/jobs	Lists all jobs.

Normal response codes: 200

6.10.1.1. Request

This operation does not accept a request body.

6.10.1.2. Response

Example 6.55. List jobs: JSON response

```
{
  "jobs": [
    {
      "is_public": false,
      "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
      "created_at": "2015-02-10 14:25:48",
      "id": "1a674c31-9aaa-4d07-b844-2bf200a1b836",
      "name": "Edp-test-job-3d60854e",
      "updated_at": null,
      "description": "",
      "interface": [],
      "libs": [
        {
          "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
          "created_at": "2015-02-10 14:25:48",
          "id": "0ff4ac10-94a4-4e25-9ac9-603afe27b100",
          "name": "binary-job-339c2d1a.jar",
          "updated_at": null,
          "description": "",
          "url": "swift://Edp-test-c71e6bce.sahara/binary-
job-339c2d1a.jar"
        }
      ],
      "type": "MapReduce",
      "mains": [],
      "is_protected": false
    },
    {
      "is_public": false,
      "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
      "created_at": "2015-02-10 14:25:44",
      "id": "4d1f3759-3497-4927-8352-910bacf24e62",
      "name": "Edp-test-job-6b6953c8",
      "updated_at": null,
      "description": "",
      "interface": [],
      "libs": [
        {
          "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
          "created_at": "2015-02-10 14:25:44",
          "id": "e0d47800-4ac1-4d63-a2e1-c92d669a44e2",
```

```
        "name": "binary-job-6f21a2f8.jar",
        "updated_at": null,
        "description": "",
        "url": "swift://Edp-test-b409ec68.sahara/binary-
job-6f21a2f8.jar"
    },
    "type": "Pig",
    "mains": [
        {
            "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
            "created_at": "2015-02-10 14:25:44",
            "id": "e073e896-f123-4b76-995f-901d786262df",
            "name": "binary-job-d4f8bd75.pig",
            "updated_at": null,
            "description": "",
            "url": "swift://Edp-test-b409ec68.sahara/binary-job-
d4f8bd75.pig"
        }
    ],
    "is_protected": false
}
]
```

6.10.2. Create job

Method	URI	Description
POST	/v1.1/{tenant_id}/jobs	Creates a job object.

Normal response codes: 202

6.10.2.1. Request

Example 6.56. Create job: JSON request

```
{
  "description": "This is pig job example",
  "mains": [
    "90d9d5ec-11aa-48bd-bc8c-34936ce0db6e"
  ],
  "libs": [
    "320a2ca7-25fd-4b48-9bc3-4fb1b6c4ff27"
  ],
  "type": "Pig",
  "name": "pig-job-example"
}
```

6.10.2.2. Response

Example 6.57. Create job: JSON response

```
{
  "job": {
    "is_public": false,
    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
    "created_at": "2015-03-27 08:48:38.630827",
    "id": "71defc8f-d005-484f-9d86-1aedf644d1ef",
    "name": "pig-job-example",
    "description": "This is pig job example",
    "interface": [],
    "libs": [
      {
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "created_at": "2015-02-10 14:25:53",
        "id": "320a2ca7-25fd-4b48-9bc3-4fb1b6c4ff27",
        "name": "binary-job",
        "updated_at": null,
        "description": "",
        "url": "internal-db://c6a925fa-ac1d-4b2e-b88a-7054e1927521"
      }
    ],
    "type": "Pig",
    "is_protected": false,
    "mains": [
      {
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "created_at": "2015-02-03 10:47:51",
        "id": "90d9d5ec-11aa-48bd-bc8c-34936ce0db6e",
        "name": "pig",
        "updated_at": null,

```



```
    "description": "",  
    "url": "internal-db://872878f6-72ea-44db-8d1d-e6a6396d2df0"  
  }  
]  
}
```

6.10.3. Show job details

Method	URI	Description
GET	/v1.1/{tenant_id}/jobs/{job_id}	Shows details for a job.

Normal response codes: 200

6.10.3.1. Request

This table shows the URI parameters for the show job details request:

Name	Type	Description
{job_id}	UUID	The UUID of the job.

This operation does not accept a request body.

6.10.3.2. Response

Example 6.58. Show job details: JSON response

```
{
  "job": {
    "is_public": false,
    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
    "created_at": "2015-02-10 14:25:48",
    "id": "1a674c31-9aaa-4d07-b844-2bf200a1b836",
    "name": "Edp-test-job",
    "updated_at": null,
    "description": "",
    "interface": [],
    "libs": [
      {
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "created_at": "2015-02-10 14:25:48",
        "id": "0ff4ac10-94a4-4e25-9ac9-603afe27b100",
        "name": "binary-job.jar",
        "updated_at": null,
        "description": "",
        "url": "swift://Edp-test-c71e6bce.sahara/binary-job.jar"
      }
    ],
    "type": "MapReduce",
    "mains": [],
    "is_protected": false
  }
}
```

6.10.4. Remove job

Method	URI	Description
DELETE	/v1.1/{tenant_id}/jobs/{job_id}	Removes a job.

Normal response codes: 204

6.10.4.1. Request

This table shows the URI parameters for the remove job request:

Name	Type	Description
{job_id}	UUID	The UUID of the job.

This operation does not accept a request body.

6.10.5. Update job object

Method	URI	Description
PATCH	/v1.1/{tenant_id}/jobs/{job_id}	Updates a job object.

Normal response codes: 202

6.10.5.1. Request

This table shows the URI parameters for the update job object request:

Name	Type	Description
{job_id}	UUID	The UUID of the job.

Example 6.59. Update job object: JSON request

```
{
  "description": "This is public pig job example",
  "name": "public-pig-job-example"
}
```

6.10.5.2. Response

Example 6.60. Update job object: JSON response

```
{
  "job": {
    "is_public": false,
    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
    "created_at": "2015-02-10 14:25:48",
    "id": "1a674c31-9aaa-4d07-b844-2bf200a1b836",
    "name": "public-pig-job-example",
    "updated_at": null,
    "description": "This is public pig job example",
    "interface": [],
    "libs": [
      {
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "created_at": "2015-02-10 14:25:48",
        "id": "0ff4ac10-94a4-4e25-9ac9-603afe27b100",
        "name": "binary-job.jar",
        "updated_at": null,
        "description": "",
        "url": "swift://Edp-test-c71e6bce.sahara/binary-job.jar"
      }
    ],
    "type": "MapReduce",
    "mains": [],
    "is_protected": false
  }
}
```

6.10.6. Run job

Method	URI	Description
POST	/v1.1/{tenant_id}/jobs/{job_id}/execute	Runs a job.

Normal response codes: 202

6.10.6.1. Request

This table shows the URI parameters for the run job request:

Name	Type	Description
{job_id}	UUID	The UUID of the job.

Example 6.61. Run job: JSON request

```
{
  "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
  "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
  "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
  "job_configs": {
    "configs": {
      "mapred.map.tasks": "1",
      "mapred.reduce.tasks": "1"
    },
    "args": [
      "arg1",
      "arg2"
    ],
    "params": {
      "param2": "value2",
      "param1": "value1"
    }
  }
}
```

6.10.6.2. Response

Example 6.62. Run job: JSON response

```
{
  "job_execution": {
    "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
    "is_protected": false,
    "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
    "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
    "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
    "created_at": "2015-09-15T09:49:24",
    "is_public": false,
    "id": "20da9edb-12ce-4b45-a473-41baeefef997",
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "job_configs": {
      "configs": {
        "mapred.reduce.tasks": "1",

```

```

        "mapred.map.tasks": "1"
      },
      "args": [
        "arg1",
        "arg2"
      ],
      "params": {
        "param2": "value2",
        "param1": "value1"
      }
    },
    "info": {
      "status": "PENDING"
    }
  }
}

```

6.11. Job executions

A job execution object represents a Hadoop job that runs on a cluster. A job execution polls the status of a running job and reports it to the user. Also a user can cancel a running job.

Method	URI	Description
GET	/v1.1/{tenant_id}/job-executions	Lists available job executions.
GET	/v1.1/{tenant_id}/job-executions/{job_execution_id}	Shows details for a job execution, by ID.
DELETE	/v1.1/{tenant_id}/job-executions/{job_execution_id}	Deletes a job execution.
PATCH	/v1.1/{tenant_id}/job-executions/{job_execution_id}	Updates a job execution.
GET	/v1.1/{tenant_id}/job-executions/{job_execution_id}/refresh-status	Refreshes the status of and shows information for a job execution.
GET	/v1.1/{tenant_id}/job-executions/{job_execution_id}/cancel	Cancels a job execution.

6.11.1. List job executions

Method	URI	Description
GET	/v1.1/{tenant_id}/job-executions	Lists available job executions.

Normal response codes: 200

6.11.1.1. Request

This operation does not accept a request body.

6.11.1.2. Response

Example 6.63. List job executions: JSON response

```
{
  "job_executions": [
    {
      "job_configs": {
        "configs": {
          "mapred.reduce.tasks": "1",
          "mapred.map.tasks": "1"
        },
        "args": [
          "arg1",
          "arg2"
        ],
        "params": {
          "param2": "value2",
          "param1": "value1"
        }
      },
      "is_protected": false,
      "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
      "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
      "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
      "created_at": "2015-09-15T09:49:24",
      "end_time": "2015-09-15T12:50:46",
      "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
      "is_public": false,
      "updated_at": "2015-09-15T09:50:46",
      "return_code": null,
      "data_source_urls": {
        "3e1bc8e6-8c69-4749-8e52-90d9341d15bc": "swift://ap-cont/
input",
        "52146b52-6540-4aac-a024-fee253cf52a9": "swift://ap-cont/
output"
      },
      "tenant_id": "808d5032ea0446889097723bfc8e919d",
      "start_time": "2015-09-15T12:49:43",
      "id": "20da9edb-12ce-4b45-a473-41baeefef997",
      "oozie_job_id": "0000001-150915094349962-oozie-hado-w",
      "info": {
        "user": "hadoop",
        "actions": [
          {
            "name": ":start:",
```

```

        "trackerUri": "-",
        "externalStatus": "OK",
        "status": "OK",
        "externalId": "-",
        "transition": "job-node",
        "data": null,
        "endTime": "Tue, 15 Sep 2015 09:49:59 GMT",
        "errorCode": null,
        "id": "0000001-150915094349962-oozie-hado-W@start:",
        "consoleUrl": "-",
        "errorMessage": null,
        "toString": "Action name[:start:] status[OK]",
        "stats": null,
        "type": ":START:",
        "retries": 0,
        "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
        "externalChildIDs": null,
        "cred": "null"
    },
    {
        "name": "job-node",
        "trackerUri": "http://172.18.168.119:8032",
        "externalStatus": "FAILED/KILLED",
        "status": "ERROR",
        "externalId": "job_1442310173665_0002",
        "transition": "fail",
        "data": null,
        "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
        "errorCode": "JA018",
        "id": "0000001-150915094349962-oozie-hado-W@job-node",
        "consoleUrl": "http://ap-cluster-all-0:8088/proxy/application_1442310173665_0002/",
        "errorMessage": "Main class [org.apache.oozie.action.hadoop.PigMain], exit code [2]",
        "toString": "Action name[job-node] status[ERROR]",
        "stats": null,
        "type": "pig",
        "retries": 0,
        "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
        "externalChildIDs": null,
        "cred": "null"
    },
    {
        "name": "fail",
        "trackerUri": "-",
        "externalStatus": "OK",
        "status": "OK",
        "externalId": "-",
        "transition": null,
        "data": null,
        "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
        "errorCode": "E0729",
        "id": "0000001-150915094349962-oozie-hado-W@fail",
        "consoleUrl": "-",
        "errorMessage": "Workflow failed, error message[Main class [org.apache.oozie.action.hadoop.PigMain], exit code [2]]",
        "toString": "Action name[fail] status[OK]",
        "stats": null,
        "type": ":KILL:",
        "retries": 0,

```



```

        "startTime": "Tue, 15 Sep 2015 09:50:17 GMT",
        "externalChildIDs": null,
        "cred": "null"
    }
],
    "createdTime": "Tue, 15 Sep 2015 09:49:58 GMT",
    "status": "KILLED",
    "group": null,
    "externalId": null,
    "acl": null,
    "run": 0,
    "appName": "job-wf",
    "parentId": null,
    "conf": "<configuration>\r\n  <property>\r\n    <name>user.name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n    <name>oozie.use.system.libpath</name>\r\n    <value>true</value>\r\n  </property>\r\n  <property>\r\n    <name>mapreduce.job.user.name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n    <name>nameNode</name>\r\n    <value>hdfs://ap-cluster-all-0:9000</value>\r\n  </property>\r\n  <property>\r\n    <name>jobTracker</name>\r\n    <value>http://172.18.168.119:8032</value>\r\n  </property>\r\n  <property>\r\n    <name>oozie.wf.application.path</name>\r\n    <value>hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml</value>\r\n  </property>\r\n</configuration>",
    "id": "00000001-150915094349962-oozie-hado-W",
    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "appPath": "hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml",
    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "toString": "Workflow id[00000001-150915094349962-oozie-hado-W]
    status[KILLED]",
    "lastModTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "consoleUrl": "http://ap-cluster-all-0.novalocal:11000/oozie?job=00000001-150915094349962-oozie-hado-W"
  }
}
]
}

```

6.11.2. Show job execution details

Method	URI	Description
GET	/v1.1/{tenant_id}/job-executions/{job_execution_id}	Shows details for a job execution, by ID.

Normal response codes: 200

6.11.2.1. Request

This table shows the URI parameters for the show job execution details request:

Name	Type	Description
{job_execution_id}	UUID	The UUID of the job execution.

This operation does not accept a request body.

6.11.2.2. Response

Example 6.64. Show job execution details: JSON response

```
{
  "job_execution": {
    "job_configs": {
      "configs": {
        "mapred.reduce.tasks": "1",
        "mapred.map.tasks": "1"
      },
      "args": [
        "arg1",
        "arg2"
      ],
      "params": {
        "param2": "value2",
        "param1": "value1"
      }
    },
    "is_protected": false,
    "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
    "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
    "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
    "created_at": "2015-09-15T09:49:24",
    "end_time": "2015-09-15T12:50:46",
    "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
    "is_public": false,
    "updated_at": "2015-09-15T09:50:46",
    "return_code": null,
    "data_source_urls": {
      "3e1bc8e6-8c69-4749-8e52-90d9341d15bc": "swift://ap-cont/input",
      "52146b52-6540-4aac-a024-fee253cf52a9": "swift://ap-cont/output"
    },
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "start_time": "2015-09-15T12:49:43",
    "id": "20da9edb-12ce-4b45-a473-41baeefef997",
    "oozie_job_id": "0000001-150915094349962-oozie-hado-W",
    "info": {
      "user": "hadoop",
```

```

"actions": [
  {
    "name": ":start:",
    "trackerUri": "-",
    "externalStatus": "OK",
    "status": "OK",
    "externalId": "-",
    "transition": "job-node",
    "data": null,
    "endTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "errorCode": null,
    "id": "0000001-150915094349962-oozie-hado-w@:start:",
    "consoleUrl": "-",
    "errorMessage": null,
    "toString": "Action name[:start:] status[OK]",
    "stats": null,
    "type": ":START:",
    "retries": 0,
    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "externalChildIDs": null,
    "cred": "null"
  },
  {
    "name": "job-node",
    "trackerUri": "http://172.18.168.119:8032",
    "externalStatus": "FAILED/KILLED",
    "status": "ERROR",
    "externalId": "job_1442310173665_0002",
    "transition": "fail",
    "data": null,
    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "errorCode": "JA018",
    "id": "0000001-150915094349962-oozie-hado-w@job-node",
    "consoleUrl": "http://ap-cluster-all-0:8088/proxy/
application_1442310173665_0002/",
    "errorMessage": "Main class [org.apache.oozie.action.
hadoop.PigMain], exit code [2]",
    "toString": "Action name[job-node] status[ERROR]",
    "stats": null,
    "type": "pig",
    "retries": 0,
    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "externalChildIDs": null,
    "cred": "null"
  },
  {
    "name": "fail",
    "trackerUri": "-",
    "externalStatus": "OK",
    "status": "OK",
    "externalId": "-",
    "transition": null,
    "data": null,
    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "errorCode": "E0729",
    "id": "0000001-150915094349962-oozie-hado-w@fail",
    "consoleUrl": "-",
    "errorMessage": "Workflow failed, error message[Main class
[org.apache.oozie.action.hadoop.PigMain], exit code [2]]",
    "toString": "Action name[fail] status[OK]",
  }
]

```

```

        "stats": null,
        "type": ":KILL:",
        "retries": 0,
        "startTime": "Tue, 15 Sep 2015 09:50:17 GMT",
        "externalChildIDs": null,
        "cred": "null"
    }
],
"createdTime": "Tue, 15 Sep 2015 09:49:58 GMT",
"status": "KILLED",
"group": null,
"externalId": null,
"acl": null,
"run": 0,
"appName": "job-wf",
"parentId": null,
"conf": "<configuration>\r\n  <property>\r\n    <name>user.
name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>
\r\n    <name>oozie.use.system.libpath</name>\r\n    <value>true</value>
\r\n  </property>\r\n  <property>\r\n    <name>mapreduce.job.user.name</
name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n
    <name>nameNode</name>\r\n    <value>hdfs://ap-cluster-all-0:9000</value>
\r\n  </property>\r\n  <property>\r\n    <name>jobTracker</name>\r\n
    <value>http://172.18.168.119:8032</value>\r\n  </property>\r\n  <property>\r
\r\n    <name>oozie.wf.application.path</name>\r\n    <value>hdfs://ap-cluster-
all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/
workflow.xml</value>\r\n  </property>\r\n</configuration>",
    "id": "00000001-150915094349962-oozie-hado-w",
    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "appPath": "hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-
example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml",
    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "toString": "Workflow id[00000001-150915094349962-oozie-hado-w]
status[KILLED]",
    "lastModTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "consoleUrl": "http://ap-cluster-all-0.novalocal:11000/oozie?job=
00000001-150915094349962-oozie-hado-w"
  }
}
}

```

6.11.3. Delete job execution

Method	URI	Description
DELETE	/v1.1/{tenant_id}/job-executions/{job_execution_id}	Deletes a job execution.

Normal response codes: 204

6.11.3.1. Request

This table shows the URI parameters for the delete job execution request:

Name	Type	Description
{job_execution_id}	UUID	The UUID of the job execution.

This operation does not accept a request body.

6.11.4. Update job execution

Method	URI	Description
PATCH	/v1.1/{tenant_id}/job-executions/{job_execution_id}	Updates a job execution.

Normal response codes: 202

6.11.4.1. Request

This table shows the URI parameters for the update job execution request:

Name	Type	Description
{job_execution_id}	UUID	The UUID of the job execution.

Example 6.65. Update job execution: JSON request

```
{
  "is_public": true
}
```

6.11.4.2. Response

Example 6.66. Update job execution: JSON response

```
{
  "job_execution": {
    "job_configs": {
      "configs": {
        "mapred.reduce.tasks": "1",
        "mapred.map.tasks": "1"
      },
      "args": [
        "arg1",
        "arg2"
      ],
      "params": {
        "param2": "value2",
        "param1": "value1"
      }
    },
    "is_protected": false,
    "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
    "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
    "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
    "created_at": "2015-09-15T09:49:24",
    "end_time": "2015-09-15T12:50:46",
    "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
    "is_public": true,
    "updated_at": "2015-09-15T09:50:46",
    "return_code": null,
    "data_source_urls": {
      "3e1bc8e6-8c69-4749-8e52-90d9341d15bc": "swift://ap-cont/input",
      "52146b52-6540-4aac-a024-fee253cf52a9": "swift://ap-cont/output"
    },
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "start_time": "2015-09-15T12:49:43",
```

```

    "id": "20da9edb-12ce-4b45-a473-41baeefef997",
    "oozie_job_id": "0000001-150915094349962-oozie-hado-W",
    "info": {
      "user": "hadoop",
      "actions": [
        {
          "name": ":start:",
          "trackerUri": "-",
          "externalStatus": "OK",
          "status": "OK",
          "externalId": "-",
          "transition": "job-node",
          "data": null,
          "endTime": "Tue, 15 Sep 2015 09:49:59 GMT",
          "errorCode": null,
          "id": "0000001-150915094349962-oozie-hado-W@:start:",
          "consoleUrl": "-",
          "errorMessage": null,
          "toString": "Action name[:start:] status[OK]",
          "stats": null,
          "type": ":START:",
          "retries": 0,
          "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
          "externalChildIDs": null,
          "cred": "null"
        },
        {
          "name": "job-node",
          "trackerUri": "http://172.18.168.119:8032",
          "externalStatus": "FAILED/KILLED",
          "status": "ERROR",
          "externalId": "job_1442310173665_0002",
          "transition": "fail",
          "data": null,
          "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
          "errorCode": "JA018",
          "id": "0000001-150915094349962-oozie-hado-W@job-node",
          "consoleUrl": "http://ap-cluster-all-0:8088/proxy/application_1442310173665_0002/",
          "errorMessage": "Main class [org.apache.oozie.action.hadoop.PigMain], exit code [2]",
          "toString": "Action name[job-node] status[ERROR]",
          "stats": null,
          "type": "pig",
          "retries": 0,
          "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
          "externalChildIDs": null,
          "cred": "null"
        },
        {
          "name": "fail",
          "trackerUri": "-",
          "externalStatus": "OK",
          "status": "OK",
          "externalId": "-",
          "transition": null,
          "data": null,
          "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
          "errorCode": "E0729",
          "id": "0000001-150915094349962-oozie-hado-W@fail",

```

```

        "consoleUrl": "-",
        "errorMessage": "Workflow failed, error message[Main class
[org.apache.oozie.action.hadoop.PigMain], exit code [2]]",
        "toString": "Action name[fail] status[OK]",
        "stats": null,
        "type": ":KILL:",
        "retries": 0,
        "startTime": "Tue, 15 Sep 2015 09:50:17 GMT",
        "externalChildIDs": null,
        "cred": "null"
    }
],
"createdTime": "Tue, 15 Sep 2015 09:49:58 GMT",
"status": "KILLED",
"group": null,
"externalId": null,
"acl": null,
"run": 0,
"appName": "job-wf",
"parentId": null,
"conf": "<configuration>\r\n  <property>\r\n    <name>user.
name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>
\r\n    <name>oozie.use.system.libpath</name>\r\n    <value>true</value>
\r\n  </property>\r\n  <property>\r\n    <name>mapreduce.job.user.name</
name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n
    <name>nameNode</name>\r\n    <value>hdfs://ap-cluster-all-0:9000</value>
\r\n  </property>\r\n  <property>\r\n    <name>jobTracker</name>\r\n
    <value>http://172.18.168.119:8032</value>\r\n  </property>\r\n  <property>\r
\n    <name>oozie.wf.application.path</name>\r\n    <value>hdfs://ap-cluster-
all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/
workflow.xml</value>\r\n  </property>\r\n</configuration>",
    "id": "0000001-150915094349962-oozie-hado-W",
    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "appPath": "hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-
example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml",
    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "toString": "Workflow id[0000001-150915094349962-oozie-hado-W]
status[KILLED]",
    "lastModTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "consoleUrl": "http://ap-cluster-all-0.novalocal:11000/oozie?job=
0000001-150915094349962-oozie-hado-W"
  }
}
}

```


6.11.5. Refresh job execution status

Method	URI	Description
GET	/v1.1/{tenant_id}/job-executions/{job_execution_id}/refresh-status	Refreshes the status of and shows information for a job execution.

Normal response codes: 200

6.11.5.1. Request

This table shows the URI parameters for the refresh job execution status request:

Name	Type	Description
{job_execution_id}	UUID	The UUID of the job execution.

This operation does not accept a request body.

6.11.5.2. Response

Example 6.67. Refresh job execution status: JSON response

```
{
  "job_execution": {
    "job_configs": {
      "configs": {
        "mapred.reduce.tasks": "1",
        "mapred.map.tasks": "1"
      },
      "args": [
        "arg1",
        "arg2"
      ],
      "params": {
        "param2": "value2",
        "param1": "value1"
      }
    },
    "is_protected": false,
    "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
    "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
    "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
    "created_at": "2015-09-15T09:49:24",
    "end_time": "2015-09-15T12:50:46",
    "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
    "is_public": false,
    "updated_at": "2015-09-15T09:50:46",
    "return_code": null,
    "data_source_urls": {
      "3e1bc8e6-8c69-4749-8e52-90d9341d15bc": "swift://ap-cont/input",
      "52146b52-6540-4aac-a024-fee253cf52a9": "swift://ap-cont/output"
    },
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "start_time": "2015-09-15T12:49:43",
    "id": "20da9edb-12ce-4b45-a473-41baeefef997",
    "oozie_job_id": "0000001-150915094349962-oozie-hado-W",
    "info": {
```

```

"user": "hadoop",
"actions": [
  {
    "name": ":start:",
    "trackerUri": "-",
    "externalStatus": "OK",
    "status": "OK",
    "externalId": "-",
    "transition": "job-node",
    "data": null,
    "endTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "errorCode": null,
    "id": "0000001-150915094349962-oozie-hado-w@:start:",
    "consoleUrl": "-",
    "errorMessage": null,
    "toString": "Action name[:start:] status[OK]",
    "stats": null,
    "type": ":START:",
    "retries": 0,
    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "externalChildIDs": null,
    "cred": "null"
  },
  {
    "name": "job-node",
    "trackerUri": "http://172.18.168.119:8032",
    "externalStatus": "FAILED/KILLED",
    "status": "ERROR",
    "externalId": "job_1442310173665_0002",
    "transition": "fail",
    "data": null,
    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "errorCode": "JA018",
    "id": "0000001-150915094349962-oozie-hado-w@job-node",
    "consoleUrl": "http://ap-cluster-all-0:8088/proxy/
application_1442310173665_0002/",
    "errorMessage": "Main class [org.apache.oozie.action.
hadoop.PigMain], exit code [2]",
    "toString": "Action name[job-node] status[ERROR]",
    "stats": null,
    "type": "pig",
    "retries": 0,
    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "externalChildIDs": null,
    "cred": "null"
  },
  {
    "name": "fail",
    "trackerUri": "-",
    "externalStatus": "OK",
    "status": "OK",
    "externalId": "-",
    "transition": null,
    "data": null,
    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "errorCode": "E0729",
    "id": "0000001-150915094349962-oozie-hado-w@fail",
    "consoleUrl": "-",
    "errorMessage": "Workflow failed, error message[Main class
[org.apache.oozie.action.hadoop.PigMain], exit code [2]]",

```

```

        "toString": "Action name[fail] status[OK]",
        "stats": null,
        "type": ":KILL:",
        "retries": 0,
        "startTime": "Tue, 15 Sep 2015 09:50:17 GMT",
        "externalChildIDs": null,
        "cred": "null"
    }
},
"createdTime": "Tue, 15 Sep 2015 09:49:58 GMT",
"status": "KILLED",
"group": null,
"externalId": null,
"acl": null,
"run": 0,
"appName": "job-wf",
"parentId": null,
"conf": "<configuration>\r\n  <property>\r\n    <name>user.
name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>
\r\n    <name>oozie.use.system.libpath</name>\r\n    <value>true</value>
\r\n  </property>\r\n  <property>\r\n    <name>mapreduce.job.user.name</
name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n
    <name>nameNode</name>\r\n    <value>hdfs://ap-cluster-all-0:9000</value>
\r\n  </property>\r\n  <property>\r\n    <name>jobTracker</name>\r\n
    <value>http://172.18.168.119:8032</value>\r\n  </property>\r\n  <property>\r
\r\n    <name>oozie.wf.application.path</name>\r\n    <value>hdfs://ap-cluster-
all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/
workflow.xml</value>\r\n  </property>\r\n</configuration>",
    "id": "00000001-150915094349962-oozie-hado-w",
    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "appPath": "hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-
example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml",
    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "toString": "Workflow id[00000001-150915094349962-oozie-hado-w]
status[KILLED]",
    "lastModTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "consoleUrl": "http://ap-cluster-all-0.novalocal:11000/oozie?job=
00000001-150915094349962-oozie-hado-w"
  }
}
}

```

6.11.6. Cancel job execution

Method	URI	Description
GET	/v1.1/{tenant_id}/job-executions/{job_execution_id}/cancel	Cancels a job execution.

Normal response codes: 200

6.11.6.1. Request

This table shows the URI parameters for the cancel job execution request:

Name	Type	Description
{job_execution_id}	UUID	The UUID of the job execution.

This operation does not accept a request body.

6.11.6.2. Response

Example 6.68. Cancel job execution: JSON response

```
{
  "job_execution": {
    "job_configs": {
      "configs": {
        "mapred.reduce.tasks": "1",
        "mapred.map.tasks": "1"
      },
      "args": [
        "arg1",
        "arg2"
      ],
      "params": {
        "param2": "value2",
        "param1": "value1"
      }
    },
    "is_protected": false,
    "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
    "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
    "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
    "created_at": "2015-09-15T09:49:24",
    "end_time": "2015-09-15T12:50:46",
    "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
    "is_public": false,
    "updated_at": "2015-09-15T09:50:46",
    "return_code": null,
    "data_source_urls": {
      "3e1bc8e6-8c69-4749-8e52-90d9341d15bc": "swift://ap-cont/input",
      "52146b52-6540-4aac-a024-fee253cf52a9": "swift://ap-cont/output"
    },
    "tenant_id": "808d5032ea0446889097723bfc8e919d",
    "start_time": "2015-09-15T12:49:43",
    "id": "20da9edb-12ce-4b45-a473-41baeefef997",
    "oozie_job_id": "0000001-150915094349962-oozie-hado-W",
    "info": {
      "user": "hadoop",

```

```

"actions": [
  {
    "name": ":start:",
    "trackerUri": "-",
    "externalStatus": "OK",
    "status": "OK",
    "externalId": "-",
    "transition": "job-node",
    "data": null,
    "endTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "errorCode": null,
    "id": "0000001-150915094349962-oozie-hado-w@:start:",
    "consoleUrl": "-",
    "errorMessage": null,
    "toString": "Action name[:start:] status[OK]",
    "stats": null,
    "type": ":START:",
    "retries": 0,
    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "externalChildIDs": null,
    "cred": "null"
  },
  {
    "name": "job-node",
    "trackerUri": "http://172.18.168.119:8032",
    "externalStatus": "FAILED/KILLED",
    "status": "ERROR",
    "externalId": "job_1442310173665_0002",
    "transition": "fail",
    "data": null,
    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "errorCode": "JA018",
    "id": "0000001-150915094349962-oozie-hado-w@job-node",
    "consoleUrl": "http://ap-cluster-all-0:8088/proxy/
application_1442310173665_0002/",
    "errorMessage": "Main class [org.apache.oozie.action.
hadoop.PigMain], exit code [2]",
    "toString": "Action name[job-node] status[ERROR]",
    "stats": null,
    "type": "pig",
    "retries": 0,
    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "externalChildIDs": null,
    "cred": "null"
  },
  {
    "name": "fail",
    "trackerUri": "-",
    "externalStatus": "OK",
    "status": "OK",
    "externalId": "-",
    "transition": null,
    "data": null,
    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "errorCode": "E0729",
    "id": "0000001-150915094349962-oozie-hado-w@fail",
    "consoleUrl": "-",
    "errorMessage": "Workflow failed, error message[Main class
[org.apache.oozie.action.hadoop.PigMain], exit code [2]]",
    "toString": "Action name[fail] status[OK]",
  }
]

```

```

        "stats": null,
        "type": ":KILL:",
        "retries": 0,
        "startTime": "Tue, 15 Sep 2015 09:50:17 GMT",
        "externalChildIDs": null,
        "cred": "null"
    }
],
"createdTime": "Tue, 15 Sep 2015 09:49:58 GMT",
"status": "KILLED",
"group": null,
"externalId": null,
"acl": null,
"run": 0,
"appName": "job-wf",
"parentId": null,
"conf": "<configuration>\r\n  <property>\r\n    <name>user.name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n    <name>oozie.use.system.libpath</name>\r\n    <value>true</value>\r\n  </property>\r\n  <property>\r\n    <name>mapreduce.job.user.name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n    <name>nameNode</name>\r\n    <value>hdfs://ap-cluster-all-0:9000</value>\r\n  </property>\r\n  <property>\r\n    <name>jobTracker</name>\r\n    <value>http://172.18.168.119:8032</value>\r\n  </property>\r\n  <property>\r\n    <name>oozie.wf.application.path</name>\r\n    <value>hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml</value>\r\n  </property>\r\n</configuration>",
    "id": "0000001-150915094349962-oozie-hado-W",
    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
    "appPath": "hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml",
    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "toString": "Workflow id[0000001-150915094349962-oozie-hado-W]
    status[KILLED]",
    "lastModTime": "Tue, 15 Sep 2015 09:50:17 GMT",
    "consoleUrl": "http://ap-cluster-all-0.novalocal:11000/oozie?job=0000001-150915094349962-oozie-hado-W"
  }
}
}

```

6.12. Job types

Each plugin that supports EDP also supports specific job types. Different versions of a plugin might actually support different job types. Configuration options vary by plugin, version, and job type.

The job types provide information about which plugins support which job types and how to configure the job types.

Method	URI	Description
GET	/v1.1/{tenant_id}/job-types	Lists all job types.

6.12.1. List job types

Method	URI	Description
GET	/v1.1/{tenant_id}/job-types	Lists all job types.

You can use query parameters to filter the response.

Normal response codes: 200

6.12.1.1. Request

This operation does not accept a request body.

6.12.1.2. Response

Example 6.69. List job types: JSON response

```
{
  "job_types": [
    {
      "plugins": [
        {
          "description": "The Apache Vanilla plugin provides
the ability to launch upstream Vanilla Apache Hadoop cluster without any
management consoles. It can also deploy the Oozie component.",
          "versions": {
            "1.2.1": {},
            "2.6.0": {}
          },
          "title": "Vanilla Apache Hadoop",
          "name": "vanilla"
        },
        {
          "description": "The Hortonworks Sahara plugin automates
the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
          "versions": {
            "1.3.2": {},
            "2.0.6": {}
          },
          "title": "Hortonworks Data Platform",
          "name": "hdp"
        },
        {
          "description": "The Cloudera Sahara plugin provides the
ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
Cloudera Manager management console.",
          "versions": {
            "5": {},
            "5.3.0": {}
          },
          "title": "Cloudera Plugin",
          "name": "cdh"
        }
      ],
      "name": "Hive"
    },
    {
```

```

        "plugins": [
            {
                "description": "The Apache Vanilla plugin provides
the ability to launch upstream Vanilla Apache Hadoop cluster without any
management consoles. It can also deploy the Oozie component.",
                "versions": {
                    "1.2.1": {},
                    "2.6.0": {}
                },
                "title": "Vanilla Apache Hadoop",
                "name": "vanilla"
            },
            {
                "description": "The Hortonworks Sahara plugin automates
the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
                "versions": {
                    "1.3.2": {},
                    "2.0.6": {}
                },
                "title": "Hortonworks Data Platform",
                "name": "hdp"
            },
            {
                "description": "The Cloudera Sahara plugin provides the
ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
Cloudera Manager management console.",
                "versions": {
                    "5": {},
                    "5.3.0": {}
                },
                "title": "Cloudera Plugin",
                "name": "cdh"
            }
        ],
        "name": "Java"
    },
    {
        "plugins": [
            {
                "description": "The Apache Vanilla plugin provides
the ability to launch upstream Vanilla Apache Hadoop cluster without any
management consoles. It can also deploy the Oozie component.",
                "versions": {
                    "1.2.1": {},
                    "2.6.0": {}
                },
                "title": "Vanilla Apache Hadoop",
                "name": "vanilla"
            },
            {
                "description": "The Hortonworks Sahara plugin automates
the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
                "versions": {
                    "1.3.2": {},
                    "2.0.6": {}
                },
                "title": "Hortonworks Data Platform",
                "name": "hdp"
            },
            {

```



```

        "description": "The Cloudera Sahara plugin provides the
ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
Cloudera Manager management console.",
        "versions": {
            "5": {},
            "5.3.0": {}
        },
        "title": "Cloudera Plugin",
        "name": "cdh"
    },
    {
        "name": "MapReduce"
    },
    {
        "plugins": [
            {
                "description": "The Apache Vanilla plugin provides
the ability to launch upstream Vanilla Apache Hadoop cluster without any
management consoles. It can also deploy the Oozie component.",
                "versions": {
                    "1.2.1": {},
                    "2.6.0": {}
                },
                "title": "Vanilla Apache Hadoop",
                "name": "vanilla"
            },
            {
                "description": "The Hortonworks Sahara plugin automates
the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
                "versions": {
                    "1.3.2": {},
                    "2.0.6": {}
                },
                "title": "Hortonworks Data Platform",
                "name": "hdp"
            },
            {
                "description": "The Cloudera Sahara plugin provides the
ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
Cloudera Manager management console.",
                "versions": {
                    "5": {},
                    "5.3.0": {}
                },
                "title": "Cloudera Plugin",
                "name": "cdh"
            }
        ],
        "name": "MapReduce.Streaming"
    },
    {
        "plugins": [
            {
                "description": "The Apache Vanilla plugin provides
the ability to launch upstream Vanilla Apache Hadoop cluster without any
management consoles. It can also deploy the Oozie component.",
                "versions": {
                    "1.2.1": {},
                    "2.6.0": {}
                },
            },

```

```

        "title": "Vanilla Apache Hadoop",
        "name": "vanilla"
    },
    {
        "description": "The Hortonworks Sahara plugin automates
the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
        "versions": {
            "1.3.2": {},
            "2.0.6": {}
        },
        "title": "Hortonworks Data Platform",
        "name": "hdp"
    },
    {
        "description": "The Cloudera Sahara plugin provides the
ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
Cloudera Manager management console.",
        "versions": {
            "5": {},
            "5.3.0": {}
        },
        "title": "Cloudera Plugin",
        "name": "cdh"
    }
],
"name": "Pig"
},
{
    "plugins": [
        {
            "description": "The Apache Vanilla plugin provides
the ability to launch upstream Vanilla Apache Hadoop cluster without any
management consoles. It can also deploy the Oozie component.",
            "versions": {
                "1.2.1": {},
                "2.6.0": {}
            },
            "title": "Vanilla Apache Hadoop",
            "name": "vanilla"
        },
        {
            "description": "The Hortonworks Sahara plugin automates
the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
            "versions": {
                "1.3.2": {},
                "2.0.6": {}
            },
            "title": "Hortonworks Data Platform",
            "name": "hdp"
        },
        {
            "description": "The Cloudera Sahara plugin provides the
ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
Cloudera Manager management console.",
            "versions": {
                "5": {},
                "5.3.0": {}
            },
            "title": "Cloudera Plugin",
            "name": "cdh"
        }
    ]
}

```

```
        }
      ],
      "name": "Shell"
    },
    {
      "plugins": [
        {
          "description": "This plugin provides an ability to launch
Spark on Hadoop CDH cluster without any management consoles.",
          "versions": {
            "1.0.0": {}
          },
          "title": "Apache Spark",
          "name": "spark"
        }
      ],
      "name": "Spark"
    }
  ]
}
```

7. Identity API v3 (CURRENT)

The Identity service generates authentication tokens that permit access to the OpenStack services REST APIs. Clients obtain this token and the URL endpoints for other service APIs by supplying their valid credentials to the authentication service.

Each time you make a REST API request to an OpenStack service, you supply your authentication token in the X-Auth-Token request header.

Like most OpenStack projects, OpenStack Identity protects its APIs by defining policy rules based on a role-based access control (RBAC) approach.

The Identity service [keystone.conf](#) configuration file sets the name and location of a JSON policy file that stores these rules.

For information about Identity API protection, see [Identity API protection with role-based access control \(RBAC\)](#) in the *OpenStack Cloud Administrator Guide*.

This page lists the Identity API operations in the following order:

- [Authentication and token management](#)
- [Credentials](#)
- [Domains](#)
- [Domain configuration](#)
- [Groups](#)
- [Policies](#)
- [Projects](#)
- [Regions](#)
- [Roles and role assignments](#)
- [Service catalog and endpoints](#)
- [Users](#)

Method	URI	Description
Authentication and token management (auth, tokens)		
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the password authentication method. Authorization is unscoped.
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the password authentication method with explicit unscoped authorization.
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the password authentication method and scopes authorization to a project or domain.
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the token authentication method. Authorization is unscoped.

Method	URI	Description
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the token authentication method and scopes authorization to a project or domain.
GET	/v3/auth/tokens	Validates and shows information for a token, including its expiration date and authorization scope.
HEAD	/v3/auth/tokens	Validates a token.
DELETE	/v3/auth/tokens	Revokes a token.
Credentials (credentials)		
POST	/v3/credentials	Creates a credential.
GET	/v3/credentials{?user_id}	Lists all credentials.
GET	/v3/credentials/{credential_id}	Shows details for a credential.
PATCH	/v3/credentials/{credential_id}	Updates a credential.
DELETE	/v3/credentials/{credential_id}	Deletes a credential.
Domains (domains, users, groups, roles)		
GET	/v3/domains{?name,enabled}	Lists all domains.
POST	/v3/domains	Creates a domain.
GET	/v3/domains/{domain_id}	Shows details for a domain.
PATCH	/v3/domains/{domain_id}	Updates a domain.
DELETE	/v3/domains/{domain_id}	Deletes a domain.
Domain configuration (domains, config) (since v3.4) (EXPERIMENTAL)		
GET	/v3/domains/{domain_id}/config	Shows details for a domain configuration.
PATCH	/v3/domains/{domain_id}/config	Updates a domain configuration.
DELETE	/v3/domains/{domain_id}/config	Deletes a domain configuration.
GET	/v3/domains/{domain_id}/config/{group}	Shows details for a domain group configuration.
PATCH	/v3/domains/{domain_id}/config/{group}	Updates a domain group configuration.
DELETE	/v3/domains/{domain_id}/config/{group}	Deletes a domain group configuration.
GET	/v3/domains/{domain_id}/config/{group}/{option}	Shows details for a domain group option configuration.
PATCH	/v3/domains/{domain_id}/config/{group}/{option}	Updates a domain group option configuration.
DELETE	/v3/domains/{domain_id}/config/{group}/{option}	Deletes a domain group option configuration.
Groups (groups, users)		
POST	/v3/groups	Creates a group.
GET	/v3/groups{?domain_id,name}	Lists groups.
GET	/v3/groups/{group_id}	Shows details for a group.
PATCH	/v3/groups/{group_id}	Updates a group.
DELETE	/v3/groups/{group_id}	Deletes a group.
GET	/v3/groups/{group_id}/users{?domain_id,description,name,enabled}	Lists the users that belong to a group.
PUT	/v3/groups/{group_id}/users/{user_id}	Adds a user to a group.
HEAD	/v3/groups/{group_id}/users/{user_id}	Validates that a user belongs to a group.

Method	URI	Description
DELETE	/v3/groups/{group_id}/users/{user_id}	Removes a user from a group.
Policies (policies)		
POST	/v3/policies	Creates a policy.
GET	/v3/policies{?type}	Lists policies.
GET	/v3/policies/{policy_id}	Shows details for a policy.
PATCH	/v3/policies/{policy_id}	Updates a policy.
DELETE	/v3/policies/{policy_id}	Deletes a policy.
Projects (projects, users, groups, roles)		
GET	/v3/projects{?domain_id,parent_id,name,enabled}	Lists projects.
POST	/v3/projects	Creates a project.
GET	/v3/projects/{project_id}	Shows details for a project.
PATCH	/v3/projects/{project_id}	Updates a project.
DELETE	/v3/projects/{project_id}	Deletes a project.
PATCH	/v3/projects/{project_id}/cascade	(Since v3.6) Enables or disables a project and its entire subtree.
DELETE	/v3/projects/{project_id}/cascade	(Since v3.6) Deletes a project and its entire subtree.
Regions (regions) (since v3.2)		
GET	/v3/regions{?parent_region_id}	Lists regions.
POST	/v3/regions	Creates a region.
GET	/v3/regions/{region_id}	Shows details for a region, by ID.
PATCH	/v3/regions/{region_id}	Updates a region.
DELETE	/v3/regions/{region_id}	Deletes a region.
Roles (roles)		
POST	/v3/roles	Creates a role.
GET	/v3/roles{?name}	Lists roles.
GET	/v3/roles/{role_id}	Shows details for a role.
PATCH	/v3/roles/{role_id}	Updates a role.
DELETE	/v3/roles/{role_id}	Deletes a role.
GET	/v3/domains/{domain_id}/users/{user_id}/roles	Lists roles for a user on a domain.
PUT	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Grants a role to a user on a domain.
HEAD	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Validates that a user has a role on a domain.
DELETE	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Revokes a role from a user on a domain.
GET	/v3/domains/{domain_id}/groups/{group_id}/roles	Lists roles for a group on a domain.
PUT	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Grants a role to a group on a domain.
HEAD	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Validates that a group has a role on a domain.
DELETE	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Revokes a role from a group on a domain.
GET	/v3/projects/{project_id}/users/{user_id}/roles	Lists roles for a user on a project.

Method	URI	Description
PUT	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Grants a role to a user on a project.
HEAD	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Validates that a user has a role on a project.
DELETE	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Revokes a role from a user on a project.
GET	/v3/projects/{project_id}/groups/{group_id}/roles	Lists roles for a group on a project.
PUT	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Grants a role to a group on a project.
HEAD	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Validates that a group has a role on a project.
DELETE	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Revokes a role from a group on a project.
GET	/v3/role_assignments{?group.id, role.id, scope.domain.id, scope.project.id, user.id, effective, include_subtree}	Lists role assignments.
Service catalog and endpoints (services, endpoints)		
GET	/v3/services{?type}	Lists all services.
POST	/v3/services	Creates a service.
GET	/v3/services/{service_id}	Shows details for a service.
PATCH	/v3/services/{service_id}	Updates a service.
DELETE	/v3/services/{service_id}	Deletes a service.
GET	/v3/endpoints{?interface, service_id}	Lists all available endpoints.
POST	/v3/endpoints	Creates an endpoint.
GET	/v3/endpoints/{endpoint_id}	Shows details for an endpoint.
PATCH	/v3/endpoints/{endpoint_id}	Updates an endpoint.
DELETE	/v3/endpoints/{endpoint_id}	Deletes an endpoint.
Users (users, groups, projects, roles)		
GET	/v3/users{?domain_id, name, enabled}	Lists users.
POST	/v3/users	Creates a user.
GET	/v3/users/{user_id}	Shows details for a user.
PATCH	/v3/users/{user_id}	Updates the password for or enables or disables a user.
DELETE	/v3/users/{user_id}	Deletes a user.
POST	/v3/users/{user_id}/password	Changes the password for a user.
GET	/v3/users/{user_id}/groups	Lists groups to which a user belongs.
GET	/v3/users/{user_id}/projects	List projects for a user.

7.1. Authentication and token management (auth, tokens)

In exchange for a set of authentication credentials, the Identity service generates tokens. A token represents the authenticated identity of a user and, optionally, grants authorization on a specific project or domain.

The body of an authentication request must include a payload that specifies the authentication method, which is password or token, the credentials, and, optionally, the authorization scope. You can scope a token to a project or domain, or the token can be unscoped. You cannot scope a token to both a project and domain.

Tokens have IDs, which the Identity API returns in the X-Subject-Token response header.

Also, validates an authentication token and lists the domains, projects, roles, and endpoints to which the token gives access. Forces the immediate revocation of a token.

After you obtain an authentication token, you can:

- Make REST API requests to other OpenStack services. You supply the ID of your authentication token in the X-Auth-Token request header.
- Validate your authentication token and list the domains, projects, roles, and endpoints that your token gives you access to.
- Use your token to request another token scoped for a different domain and project.
- Force the immediate revocation of a token.
- List revoked public key infrastructure (PKI) tokens.

The Identity API treats expired tokens as no longer valid tokens. The deployment determines how long expired tokens are stored.

These authentication errors can occur:

Table 7.1. Authentication errors

Response code	Description
Bad Request (400)	The Identity service failed to parse the request as expected. One of the following errors occurred: • A required attribute was missing. • An attribute that is not allowed was specified, such as an ID on a POST request in a basic CRUD operation. • An attribute of an unexpected data type was specified.
Unauthorized (401)	One of the following errors occurred: • Authentication was not performed. • The specified X-Auth-Token header is not valid. • The authentication credentials are not valid.
Forbidden (403)	The identity was successfully authenticated but it is not authorized to perform the requested action.
Not Found (404)	An operation failed because a referenced entity cannot be found by ID. For a POST request, the referenced entity might be specified in the request body rather than in the resource path.
Conflict (409)	A POST or PATCH operation failed. For example, a client tried to update a unique attribute for an entity, which conflicts with that of another entity in the same collection. Or, a client issued a create operation twice on a collection with a user-defined, unique attribute. For example, a client made a POST /users request two times for the unique, user-defined name attribute for a user entity.

Method	URI	Description
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the password authentication method. Authorization is unscoped.
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the password authentication method with explicit unscoped authorization.
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the password authentication method and scopes authorization to a project or domain.
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the token authentication method. Authorization is unscoped.
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the token authentication method and scopes authorization to a project or domain.
GET	/v3/auth/tokens	Validates and shows information for a token, including its expiration date and authorization scope.
HEAD	/v3/auth/tokens	Validates a token.
DELETE	/v3/auth/tokens	Revokes a token.

7.1.1. Password authentication with unscoped authorization

Method	URI	Description
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the password authentication method. Authorization is unscoped.

The request body must include a payload that specifies the authentication method, which is password, and the user, by ID or name, and password credentials.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.1.1.1. Request

This table shows the query parameters for the password authentication with unscoped authorization request:

Name	Type	Description
nocatalog	String (Optional)	(Since v3.1) The authentication response excludes the service catalog. By default, the response includes the service catalog.

Example 7.1. Password authentication with unscoped authorization: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "id": "423f19a4ac1e4f48bbb4180756e6eb6c",
          "password": "devstacker"
        }
      }
    }
  }
}
```

Example 7.2. Password authentication with unscoped authorization: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ]
    }
  }
}
```

```
    ],
    "password": {
      "user": {
        "name": "admin",
        "domain": {
          "id": "default"
        },
      },
      "password": "devstacker"
    }
  }
}
```

7.1.1.2. Response

Example 7.3. Password authentication with unscoped authorization: JSON response

```
{
  "token": {
    "methods": [
      "password"
    ],
    "expires_at": "2015-11-06T15:32:17.893769Z",
    "extras": {},
    "user": {
      "domain": {
        "id": "default",
        "name": "Default"
      },
      "id": "423f19a4ac1e4f48bbb4180756e6eb6c",
      "name": "admin"
    },
    "audit_ids": [
      "ZzZwkUflQfygX7pdYDBCQQ"
    ],
    "issued_at": "2015-11-06T14:32:17.893797Z"
  }
}
```

7.1.2. Password authentication with explicit unscoped authorization

Method	URI	Description
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the password authentication method with explicit unscoped authorization.

The request body must include a payload that specifies the password authentication method, the credentials, and the unscoped authorization scope.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.1.2.1. Request

This table shows the query parameters for the password authentication with explicit unscoped authorization request:

Name	Type	Description
nocatalog	String (Optional)	(Since v3.1) The authentication response excludes the service catalog. By default, the response includes the service catalog.

Example 7.4. Password authentication with explicit unscoped authorization: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "id": "ee4dfb6e5540447cb3741905149d9b6e",
          "password": "devstacker"
        }
      }
    },
    "scope": "unscoped"
  }
}
```

7.1.2.2. Response

Example 7.5. Password authentication with explicit unscoped authorization: JSON response

```
{
```

```
"token": {
  "methods": [
    "password"
  ],
  "expires_at": "2015-11-09T01:42:57.527363Z",
  "extras": {},
  "user": {
    "domain": {
      "id": "default",
      "name": "Default"
    },
    "id": "ee4dfb6e5540447cb3741905149d9b6e",
    "name": "admin"
  },
  "audit_ids": [
    "lC2Wj1jbQe-dLjLy0x4qPQ"
  ],
  "issued_at": "2015-11-09T00:42:57.527404Z"
}
```

7.1.3. Password authentication with scoped authorization

Method	URI	Description
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the password authentication method and scopes authorization to a project or domain.

The request body must include a payload that specifies the password authentication method, the credentials, and the project or domain authorization scope.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.1.3.1. Request

This table shows the query parameters for the password authentication with scoped authorization request:

Name	Type	Description
nocatalog	String (Optional)	(Since v3.1) The authentication response excludes the service catalog. By default, the response includes the service catalog.

Example 7.6. Password authentication with scoped authorization: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "id": "ee4dfb6e5540447cb3741905149d9b6e",
          "password": "devstack"
        }
      }
    },
    "scope": {
      "project": {
        "id": "a6944d763bf64ee6a275f1263fae0352"
      }
    }
  }
}
```

7.1.3.2. Response

Example 7.7. Password authentication with scoped authorization: JSON response

```
{
```

```

"token": {
  "methods": [
    "password"
  ],
  "roles": [
    {
      "id": "51cc68287d524c759f47c811e6463340",
      "name": "admin"
    }
  ],
  "expires_at": "2015-11-07T02:58:43.578887Z",
  "project": {
    "domain": {
      "id": "default",
      "name": "Default"
    },
    "id": "a6944d763bf64ee6a275f1263fae0352",
    "name": "admin"
  },
  "catalog": [
    {
      "endpoints": [
        {
          "region_id": "RegionOne",
          "url": "http://23.253.248.171:5000/v2.0",
          "region": "RegionOne",
          "interface": "public",
          "id": "068d1b359ee84b438266cb736d81de97"
        },
        {
          "region_id": "RegionOne",
          "url": "http://23.253.248.171:35357/v2.0",
          "region": "RegionOne",
          "interface": "admin",
          "id": "8bfc846841ab441ca38471be6d164ced"
        },
        {
          "region_id": "RegionOne",
          "url": "http://23.253.248.171:5000/v2.0",
          "region": "RegionOne",
          "interface": "internal",
          "id": "beb6d358c3654b4bada04d4663b640b9"
        }
      ],
      "type": "identity",
      "id": "050726f278654128aba89757ae25950c",
      "name": "keystone"
    },
    {
      "endpoints": [
        {
          "region_id": "RegionOne",
          "url": "http://23.253.248.171:8774/v2/
a6944d763bf64ee6a275f1263fae0352",
          "region": "RegionOne",
          "interface": "admin",
          "id": "ae36c0dbb0634e1dbf711f9fc2359975"
        },
        {
          "region_id": "RegionOne",

```

```

        "url": "http://23.253.248.171:8774/v2/
a6944d763bf64ee6a275f1263fae0352",
        "region": "RegionOne",
        "interface": "internal",
        "id": "d286b51530144d90a4de52d214d3ad1e"
    },
    {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8774/v2/
a6944d763bf64ee6a275f1263fae0352",
        "region": "RegionOne",
        "interface": "public",
        "id": "d6e681dd4aab4ae5a0937ed60bb4ae33"
    }
],
"type": "compute_legacy",
"id": "1c4bfbabe3b346b1bbe27a4b3258964f",
"name": "nova_legacy"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8776/v2/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "internal",
            "id": "2dce7429526e44808235fe918063a914"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8776/v2/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "public",
            "id": "a9a9929e6dc645c882ac1abd8bf73d38"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8776/v2/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "admin",
            "id": "c7d5f958df7f4c8da84db91094bdc198"
        }
    ],
    "type": "volume_v2",
    "id": "202382a1b8a94210bb3120af958092c4",
    "name": "cinderv2"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8080",
            "region": "RegionOne",
            "interface": "admin",
            "id": "29b58f1406804c8180ccc01793ff8038"
        }
    ],
    {

```



```

        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8080/v1/
AUTH_a6944d763bf64ee6a275f1263fae0352",
        "region": "RegionOne",
        "interface": "public",
        "id": "4c2c2968008c4e77973a5922e192d982"
    },
    {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8080/v1/
AUTH_a6944d763bf64ee6a275f1263fae0352",
        "region": "RegionOne",
        "interface": "internal",
        "id": "f6e7b28008bf41eaa114176a15ac1410"
    }
],
"type": "object-store",
"id": "52fecdef9ad543779c1312392cc2b115",
"name": "swift"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9696/",
            "region": "RegionOne",
            "interface": "public",
            "id": "6a2840dc63bf433592cd8bca2183eb3c"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9696/",
            "region": "RegionOne",
            "interface": "internal",
            "id": "7967cf45f8ab439a80cf24420e5ffdf0e"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9696/",
            "region": "RegionOne",
            "interface": "admin",
            "id": "84943ce595264303bd44e5d6d79bea7b"
        }
    ],
    "type": "network",
    "id": "67b993549db94296a853d635b48db3c9",
    "name": "neutron"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8888",
            "region": "RegionOne",
            "interface": "public",
            "id": "2896609ef89741148bbd8c93babf5a12"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8888",

```

```

        "region": "RegionOne",
        "interface": "internal",
        "id": "30de385478fe4325849f98d1e45bc5e6"
    },
    {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8888",
        "region": "RegionOne",
        "interface": "admin",
        "id": "41256dc4b3c942daa383f940a9a56536"
    }
],
"type": "messaging",
"id": "6fc9cc3e6b3843b899478554f9e297d3",
"name": "zaqar"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9000",
            "region": "RegionOne",
            "interface": "admin",
            "id": "07ea5fe3ae784001a73f131fb1764bf4"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9000",
            "region": "RegionOne",
            "interface": "internal",
            "id": "31e709ecb15d4881806dbced4eb3e60e"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9000",
            "region": "RegionOne",
            "interface": "public",
            "id": "a0c2a150a6ae4bbc85f1d428b9d78a1b"
        }
    ],
    "type": "messaging-websocket",
    "id": "816031f798cc4ac7879eda0cf9cf033a",
    "name": "zaqar-websocket"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8773/",
            "region": "RegionOne",
            "interface": "internal",
            "id": "24df0277c2b6499ea6051bea8c59ff74"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8773/",
            "region": "RegionOne",
            "interface": "public",
            "id": "438f4b3f3c314bbf988f1442cc3ddfa5"
        }
    ],

```

```

        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8773/",
            "region": "RegionOne",
            "interface": "admin",
            "id": "90a1c9fab54c452fa02a59ff87165029"
        }
    ],
    "type": "ec2",
    "id": "915e2a8b1f314d55bba28432c9d5c1de",
    "name": "ec2"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8774/v2.1/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "admin",
            "id": "2511589f262a407bb0071a814a480af4"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8774/v2.1/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "internal",
            "id": "9cf9209ae4fc4673a7295611001cf0ae"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8774/v2.1/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "public",
            "id": "d200b2509e1343e3887dcc465b4fa534"
        }
    ],
    "type": "compute",
    "id": "a226b3eeb5594f50bf8b6df94636ed28",
    "name": "nova"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8004/v1/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "internal",
            "id": "bf2fe80c2a614e438d3e55b00e85b9fff"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8004/v1/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "admin",
            "id": "bfc9615fc24e4045aaf719f060984bf1"
        }
    ]
}

```

```

        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8004/v1/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "public",
            "id": "c76cf9930b0f4ccea6b1157f80119cfc"
        }
    ],
    "type": "orchestration",
    "id": "a5f7070bda40443fa3819fbdf1689af1",
    "name": "heat"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8776/v1/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "internal",
            "id": "3e321c2c6fa04152b3e86c18b91b93ae"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8776/v1/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "admin",
            "id": "55aef0f2557449d4946dc9461b73a63b"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8776/v1/
a6944d763bf64ee6a275f1263fae0352",
            "region": "RegionOne",
            "interface": "public",
            "id": "7c91a66a200e458ca6e4e00fddf4d98b"
        }
    ],
    "type": "volume",
    "id": "b6b5edc3fc384b6787149e91b3b31988",
    "name": "cinder"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9292",
            "region": "RegionOne",
            "interface": "public",
            "id": "512c10d230874ad295662157eeab0135"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9292",
            "region": "RegionOne",
            "interface": "internal",
            "id": "614b1ab241da47a8b3a4e8f67b771446"
        }
    ]
}

```

```

        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9292",
            "region": "RegionOne",
            "interface": "admin",
            "id": "9cef78a4286c42f3b977fbe4d5f927a6"
        }
    ],
    "type": "image",
    "id": "d512f8860c0f45cf99b1c3cef86cfd97",
    "name": "glance"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8000/v1",
            "region": "RegionOne",
            "interface": "internal",
            "id": "7f776d5a83d346b48e519555362b1da6"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8000/v1",
            "region": "RegionOne",
            "interface": "public",
            "id": "8303a7225a2d439fa39905c6a20202c3"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8000/v1",
            "region": "RegionOne",
            "interface": "admin",
            "id": "942fa998d1c644e0b0c085d5a0995a13"
        }
    ],
    "type": "cloudformation",
    "id": "ed0805af6ee54a19ad7e5add8465ac41",
    "name": "heat-cfn"
}
],
"extras": {},
"user": {
    "domain": {
        "id": "default",
        "name": "Default"
    },
    "id": "ee4dfb6e5540447cb3741905149d9b6e",
    "name": "admin"
},
"audit_ids": [
    "3T2dc1CGQxyJsHdDu1xkcw"
],
"issued_at": "2015-11-07T01:58:43.578929Z"
}
}

```

7.1.4. Token authentication with unscoped authorization

Method	URI	Description
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the token authentication method. Authorization is unscoped.

In the request body, provide the token ID.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.1.4.1. Request

This table shows the query parameters for the token authentication with unscoped authorization request:

Name	Type	Description
nocatalog	String (Optional)	(Since v3.1) The authentication response excludes the service catalog. By default, the response includes the service catalog.

Example 7.8. Token authentication with unscoped authorization: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "token"
      ],
      "token": {
        "id": "'$OS_TOKEN'"
      }
    }
  }
}
```

7.1.4.2. Response

Example 7.9. Token authentication with unscoped authorization: JSON response

```
{
  "token": {
    "methods": [
      "token"
    ],
    "expires_at": "2015-11-05T22:00:11.000000Z",
    "extras": {},
    "user": {
      "domain": {
```

```
        "id": "default",
        "name": "Default"
    },
    "id": "10a2e6e717a245d9acad3e5f97aeca3d",
    "name": "admin"
},
"audit_ids": [
    "mAjXQhiYRyKwkB4qygdLVg"
],
"issued_at": "2015-11-05T21:00:33.819948Z"
}
```

7.1.5. Token authentication with scoped authorization

Method	URI	Description
POST	/v3/auth/tokens{?nocatalog}	Authenticates an identity and generates a token. Uses the token authentication method and scopes authorization to a project or domain.

In the request body, provide the token ID and the project or domain authorization scope.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.1.5.1. Request

This table shows the query parameters for the token authentication with scoped authorization request:

Name	Type	Description
nocatalog	String (Optional)	(Since v3.1) The authentication response excludes the service catalog. By default, the response includes the service catalog.

Example 7.10. Token authentication with scoped authorization: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "token"
      ],
      "token": {
        "id": "'$OS_TOKEN'"
      }
    },
    "scope": {
      "project": {
        "id": "5b50efd009b540559104ee3c03bbb2b7"
      }
    }
  }
}
```

7.1.5.2. Response

Example 7.11. Token authentication with scoped authorization: JSON response

```
{
  "token": {
    "methods": [
      "token"
    ],
    "roles": [
```



```

        {
            "id": "5090055d6bd547dc83e0e8f070803708",
            "name": "admin"
        }
    ],
    "expires_at": "2015-11-05T22:00:11.000000Z",
    "project": {
        "domain": {
            "id": "default",
            "name": "Default"
        },
        "id": "5b50efd009b540559104ee3c03bbb2b7",
        "name": "admin"
    },
    "catalog": [
        {
            "endpoints": [
                {
                    "region_id": "RegionOne",
                    "url": "http://23.253.248.171:9292",
                    "region": "RegionOne",
                    "interface": "admin",
                    "id": "b2605da9b25943beb49b2bd86aca2202"
                },
                {
                    "region_id": "RegionOne",
                    "url": "http://23.253.248.171:9292",
                    "region": "RegionOne",
                    "interface": "public",
                    "id": "c4d1184caf8c4351bff4bf502a09684e"
                },
                {
                    "region_id": "RegionOne",
                    "url": "http://23.253.248.171:9292",
                    "region": "RegionOne",
                    "interface": "internal",
                    "id": "cd73bda89e3948738c2721a8c3acac54"
                }
            ],
            "type": "image",
            "id": "495df2483dc145dbb6b34bfbdd787aae",
            "name": "glance"
        },
        {
            "endpoints": [
                {
                    "region_id": "RegionOne",
                    "url": "http://23.253.248.171:8773/",
                    "region": "RegionOne",
                    "interface": "internal",
                    "id": "7d03218a7f4246e8b9e3992318bf5397"
                },
                {
                    "region_id": "RegionOne",
                    "url": "http://23.253.248.171:8773/",
                    "region": "RegionOne",
                    "interface": "public",
                    "id": "9ad7f8ce438c4212b8aac930bca04c86"
                }
            ],

```

```

        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8773/",
        "region": "RegionOne",
        "interface": "admin",
        "id": "d84aad1a45c44e4da09b719167383049"
    },
    ],
    "type": "ec2",
    "id": "54204024bb7d4665a8efc34fc758f1f7",
    "name": "ec2"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9000",
            "region": "RegionOne",
            "interface": "admin",
            "id": "1077687c18514490a3ec980eadd1bd13"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9000",
            "region": "RegionOne",
            "interface": "public",
            "id": "1e86d8bef1514c3fba8d157a22ccce88"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9000",
            "region": "RegionOne",
            "interface": "internal",
            "id": "f6a6b7bbba66443ead3a0e31a008c271"
        }
    ],
    "type": "messaging-websocket",
    "id": "6b8655af7d044a15bec3cdca4f2919f8",
    "name": "zaqar-websocket"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8004/v1/5b50efd009b540559104ee3c03bbb2b7",
            "region": "RegionOne",
            "interface": "admin",
            "id": "083663fd231e40ad97384ad3efb9f1b7"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8004/v1/5b50efd009b540559104ee3c03bbb2b7",
            "region": "RegionOne",
            "interface": "internal",
            "id": "0f4b7054ea27450eac43f685a4fc1d2c"
        },
        {
            "region_id": "RegionOne",

```

```

        "url": "http://23.253.248.171:8004/v1/
5b50efd009b540559104ee3c03bbb2b7",
        "region": "RegionOne",
        "interface": "public",
        "id": "5f3ea39df2e44378b1802a1a87ef9ac4"
    },
    ],
    "type": "orchestration",
    "id": "6d6346ff2ca842e5968373fbb93e231f",
    "name": "heat"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8774/v2.1/
5b50efd009b540559104ee3c03bbb2b7",
            "region": "RegionOne",
            "interface": "public",
            "id": "bc2230a70d6a444e9fba75b85fbda41b"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8774/v2.1/
5b50efd009b540559104ee3c03bbb2b7",
            "region": "RegionOne",
            "interface": "internal",
            "id": "d8102dc2b9984d04b30b91b0a6037470"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8774/v2.1/
5b50efd009b540559104ee3c03bbb2b7",
            "region": "RegionOne",
            "interface": "admin",
            "id": "f8253a53edd749bf8b107a53a5d47a82"
        }
    ],
    "type": "compute",
    "id": "75df965385cc4120a17110c1fde00182",
    "name": "nova"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:35357/v2.0",
            "region": "RegionOne",
            "interface": "admin",
            "id": "0ceeb58592274caea5bc942a07d5473f"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:5000/v2.0",
            "region": "RegionOne",
            "interface": "internal",
            "id": "8126f2c7021d413e9c98ec3a0ba0fd58"
        },
        {
            "region_id": "RegionOne",

```

```

        "url": "http://23.253.248.171:5000/v2.0",
        "region": "RegionOne",
        "interface": "public",
        "id": "c693879254544e3fb502e795a3f6acc8"
    },
    {
        "type": "identity",
        "id": "78aad571d38049e69c866c2abac76af6",
        "name": "keystone"
    },
    {
        "endpoints": [
            {
                "region_id": "RegionOne",
                "url": "http://23.253.248.171:8776/v1/5b50efd009b540559104ee3c03bbb2b7",
                "region": "RegionOne",
                "interface": "admin",
                "id": "3654138dc64a45aeb5a8153f2a089c74"
            },
            {
                "region_id": "RegionOne",
                "url": "http://23.253.248.171:8776/v1/5b50efd009b540559104ee3c03bbb2b7",
                "region": "RegionOne",
                "interface": "internal",
                "id": "7a0d12d0b731afd9b53d1618ab546ea"
            },
            {
                "region_id": "RegionOne",
                "url": "http://23.253.248.171:8776/v1/5b50efd009b540559104ee3c03bbb2b7",
                "region": "RegionOne",
                "interface": "public",
                "id": "82b68ff3aedb43e2acc8307234d3fd0b"
            }
        ],
        "type": "volume",
        "id": "80491007c0ab462daaa9087250325f59",
        "name": "cinder"
    },
    {
        "endpoints": [
            {
                "region_id": "RegionOne",
                "url": "http://23.253.248.171:8000/v1",
                "region": "RegionOne",
                "interface": "internal",
                "id": "24dfa252fba64469b8b1a832f04bded9"
            },
            {
                "region_id": "RegionOne",
                "url": "http://23.253.248.171:8000/v1",
                "region": "RegionOne",
                "interface": "public",
                "id": "e0a01d6cd3be4f6abcc72367b2d87993"
            },
            {
                "region_id": "RegionOne",
                "url": "http://23.253.248.171:8000/v1",

```

```

        "region": "RegionOne",
        "interface": "admin",
        "id": "f33f79d42df247e1bf6daf43a548b014"
    },
    ],
    "type": "cloudformation",
    "id": "ac5cc6e3c62840818ab338c981d5603f",
    "name": "heat-cfn"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9696/",
            "region": "RegionOne",
            "interface": "admin",
            "id": "3e78c357b3c8469fbea12eb681f88a0c"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9696/",
            "region": "RegionOne",
            "interface": "public",
            "id": "89d2aad3dc8e478fbabb21dd7db0962a"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:9696/",
            "region": "RegionOne",
            "interface": "internal",
            "id": "b6d4a8cf5e4042848a749a3116497e55"
        }
    ],
    "type": "network",
    "id": "b33660edd1eb45e485f7e5f14401a739",
    "name": "neutron"
},
{
    "endpoints": [
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8888",
            "region": "RegionOne",
            "interface": "public",
            "id": "1f8287cf963948778ab0eb109d9f857d"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8888",
            "region": "RegionOne",
            "interface": "internal",
            "id": "3adf5f9cc5184d92af5ff0fdef043e4a"
        },
        {
            "region_id": "RegionOne",
            "url": "http://23.253.248.171:8888",
            "region": "RegionOne",
            "interface": "admin",
            "id": "f747223060b3414f947fdcdca2ce8714"
        }
    ]
}

```

```

    ],
    "type": "messaging",
    "id": "cf3e38e9aed54e2d84ea64485317d7a0",
    "name": "zaqar"
  },
  {
    "endpoints": [
      {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8774/v2/5b50efd009b540559104ee3c03bbb2b7",
        "region": "RegionOne",
        "interface": "public",
        "id": "08f507ccb552476b98f3af7718f25557"
      },
      {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8774/v2/5b50efd009b540559104ee3c03bbb2b7",
        "region": "RegionOne",
        "interface": "admin",
        "id": "d20091ba591347b2b419e5fbde9b7976"
      },
      {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8774/v2/5b50efd009b540559104ee3c03bbb2b7",
        "region": "RegionOne",
        "interface": "internal",
        "id": "e6b667776e7245dea6e39f2820e080b0"
      }
    ],
    "type": "compute_legacy",
    "id": "d442e96b273a48018567aeec5800c3e0",
    "name": "nova_legacy"
  },
  {
    "endpoints": [
      {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8776/v2/5b50efd009b540559104ee3c03bbb2b7",
        "region": "RegionOne",
        "interface": "internal",
        "id": "012c78a6694a494995c58d5955fb7822"
      },
      {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8776/v2/5b50efd009b540559104ee3c03bbb2b7",
        "region": "RegionOne",
        "interface": "admin",
        "id": "802d5de210874f068ba31c7e27c29d70"
      },
      {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8776/v2/5b50efd009b540559104ee3c03bbb2b7",
        "region": "RegionOne",
        "interface": "public",

```

```

        "id": "b37ada66e02e44c9a9a7976d77365503"
      }
    ],
    "type": "volumev2",
    "id": "d93e78c7967f49acbdd732b9dd97e0d0",
    "name": "cinderv2"
  },
  {
    "endpoints": [
      {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8080/v1/
AUTH_5b50efd009b540559104ee3c03bbb2b7",
        "region": "RegionOne",
        "interface": "public",
        "id": "265ce88a0e1642fc90b2ec20ccb279ff"
      },
      {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8080",
        "region": "RegionOne",
        "interface": "admin",
        "id": "500b7f066d39492faff8a3f710fb5a2f"
      },
      {
        "region_id": "RegionOne",
        "url": "http://23.253.248.171:8080/v1/
AUTH_5b50efd009b540559104ee3c03bbb2b7",
        "region": "RegionOne",
        "interface": "internal",
        "id": "a33b0684f817405280df1f5600777a75"
      }
    ],
    "type": "object-store",
    "id": "da1b1b5c529946fcb3ee3abdcf376fcb",
    "name": "swift"
  }
],
"extras": {},
"user": {
  "domain": {
    "id": "default",
    "name": "Default"
  },
  "id": "10a2e6e717a245d9acad3e5f97aeca3d",
  "name": "admin"
},
"audit_ids": [
  "wLc7nDMsQiKqf8VFU4ySpg"
],
"issued_at": "2015-11-05T21:32:30.505384Z"
}
}

```

7.1.6. Validate and show information for token

Method	URI	Description
GET	/v3/auth/tokens	Validates and shows information for a token, including its expiration date and authorization scope.

Pass your own token in the X-Auth-Token request header.

Pass the token that you want to validate in the X-Subject-Token request header.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.1.6.1. Request

This table shows the header parameters for the validate and show information for token request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.
X-Subject-Token	String (Required)	The authentication token for which you want to perform the operation.

Example 7.12. Validate and show information for token: JSON request

Headers:

X-Auth-Token: 1dd7e3

X-Subject-Token: c67580

This operation does not accept a request body.

7.1.6.2. Response

Example 7.13. Validate and show information for token: JSON response

```
{
  "token": {
    "methods": [
      "token"
    ],
    "expires_at": "2015-11-05T22:00:11.000000Z",
    "extras": {},
    "user": {
      "domain": {
        "id": "default",
        "name": "Default"
      },
      "id": "10a2e6e717a245d9acad3e5f97aeca3d",
```



```
        "name": "admin"
      },
      "audit_ids": [
        "mAjXQhiYRyKwkB4qygdLVg"
      ],
      "issued_at": "2015-11-05T21:00:33.819948Z"
    }
  }
```

7.1.7. Check token

Method	URI	Description
HEAD	/v3/auth/tokens	Validates a token.

This call is similar to GET /auth/tokens but no response body is provided even in the X-Subject-Token header.



Important

The Identity API returns the same response as when the subject token was issued by POST /auth/tokens even if an error occurs because the token is not valid. An HTTP 204 response code indicates that the X-Subject-Token is valid.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.1.7.1. Request

This table shows the header parameters for the check token request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.
X-Subject-Token	String (Required)	The authentication token for which you want to perform the operation.

Example 7.14. Check token: JSON request

Headers:

X-Auth-Token: 1dd7e3

X-Subject-Token: c67580

This operation does not accept a request body.

7.1.7.2. Response

This table shows the header parameters for the check token response:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.
X-Subject-Token	String (Required)	The authentication token. An authentication response returns the token ID in this header rather than in the response body.

7.1.8. Revoke token

Method	URI	Description
DELETE	/v3/auth/tokens	Revokes a token.

This call is similar to the **HEAD** /auth/tokens call except that the X-Subject-Token token is immediately not valid, regardless of the expires_at attribute value. An additional X-Auth-Token is not required.

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.1.8.1. Request

This table shows the header parameters for the revoke token request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.
X-Subject-Token	String (Required)	The authentication token for which you want to perform the operation.

Example 7.15. Revoke token: JSON request

Headers:

X-Auth-Token: 1dd7e3

X-Subject-Token: c67580

This operation does not accept a request body.

7.2. Credentials (credentials)

In exchange for a set of authentication credentials that the user submits, the Identity service generates and returns a token. A token represents the authenticated identity of a user and, optionally, grants authorization on a specific project or domain.

You can list all credentials, and create, show details for, update, and delete a credential.

Method	URI	Description
POST	/v3/credentials	Creates a credential.
GET	/v3/credentials{?user_id}	Lists all credentials.
GET	/v3/credentials/{credential_id}	Shows details for a credential.
PATCH	/v3/credentials/{credential_id}	Updates a credential.
DELETE	/v3/credentials/{credential_id}	Deletes a credential.

7.2.1. Create credential

Method	URI	Description
POST	/v3/credentials	Creates a credential.

The following example shows how to create an EC2-style credential. The credential blob is a string that contains a JSON-serialized dictionary with the access and secret keys. This format is required when you specify the `ec2` type. To specify other credentials, such as `access_key`, change the type and contents of the data blob.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.2.1.1. Request

Example 7.16. Create credential: JSON request

```
{
  "credential": {
    "blob": "{\"access\":\"181920\", \"secret\":\"secretKey\"}",
    "project_id": "731fc6f265cd486d900f16e84c5cb594",
    "type": "ec2",
    "user_id": "bb5476fd12884539b41d5a88f838d773"
  }
}
```

7.2.1.2. Response

Example 7.17. Create credential: JSON response

```
{
  "credential": {
    "user_id": "bb5476fd12884539b41d5a88f838d773",
    "links": {
      "self": "http://localhost:5000/v3/credentials/3d3367228f9c7665266604462ec60029bcd83ad89614021a80b2eb879c572510"
    },
    "blob": "{\"access\":\"181920\", \"secret\":\"secretKey\"}",
    "project_id": "731fc6f265cd486d900f16e84c5cb594",
    "type": "ec2",
    "id": "3d3367228f9c7665266604462ec60029bcd83ad89614021a80b2eb879c572510"
  }
}
```

7.2.2. List credentials

Method	URI	Description
GET	/v3/credentials{?user_id}	Lists all credentials.

Optionally, you can include the `user_id` query parameter in the URI to filter the response by a user.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.2.2.1. Request

This table shows the query parameters for the list credentials request:

Name	Type	Description
user_id	UUID (Optional)	Filters the response by a user ID.

This operation does not accept a request body.

7.2.2.2. Response

Example 7.18. List credentials: JSON response

```
{
  "credentials": [
    {
      "user_id": "bb5476fd12884539b41d5a88f838d773",
      "links": {
        "self": "http://localhost:5000/v3/credentials/207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7"
      },
      "blob": "{\"access\": \"a42a27755ce6442596b049bd7dd8a563\", \"secret\": \"71faf1d40bb24c82b479b1c6fbbd9f0c\", \"trust_id\": null}\", \"project_id\": \"6e01855f345f4c59812999b5e459137d\", \"type\": \"ec2\", \"id\": \"207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7\""}
    },
    {
      "user_id": "6f556708d04b4ea6bc72d7df2296b71a",
      "links": {
        "self": "http://localhost:5000/v3/credentials/2441494e52ab6d594a34d74586075cb299489bdd1e9389e3ab06467a4f460609"
      },
      "blob": "{\"access\": \"7da79ff0aa364e1396f067e352b9b79a\", \"secret\": \"7a18d68ba8834b799d396f3ff6f1e98c\", \"trust_id\": null}\", \"project_id\": \"1a1d14690f3c4ec5bf5f321c5fde3c16\", \"type\": \"ec2\", \"id\": \"2441494e52ab6d594a34d74586075cb299489bdd1e9389e3ab06467a4f460609\""}
    }
  ]
}
```

```

    {
      "user_id": "c14107e65d5c4a7f8894fc4b3fc209ff",
      "links": {
        "self": "http://localhost:5000/v3/credentials/
3397b204b5f04c495bcd8f34c8a39996f280f9172658241873e15f070ec79d7"
      },
      "blob": "{\"access\": \"db9c58a558534a10a070110de4f9f20c\", \"
\"secret\": \"973e790b88db447ba6f93bca02bc745b\", \"trust_id\": null}\",
      "project_id": "7396e43183db40dcbf40dd727637b548",
      "type": "ec2",
      "id":
      "3397b204b5f04c495bcd8f34c8a39996f280f9172658241873e15f070ec79d7"
    },
    {
      "user_id": "915cc5f8cca6466aba6c6be06cbabfdf",
      "links": {
        "self": "http://localhost:5000/v3/credentials/
352d5dd7a4aa19c4f2f23ee288bf65dc23a0bc293f40ffd2128ffe6a8cf3e871"
      },
      "blob": "{\"access\": \"817c6c3487a440c1a0b1d3f92b30ca37\", \"
\"secret\": \"47d681117d1c46e69a0c9ec811dae2e9\", \"trust_id\": null}\",
      "project_id": "2bf9767f9db949ee8364262a28a23062",
      "type": "ec2",
      "id":
      "352d5dd7a4aa19c4f2f23ee288bf65dc23a0bc293f40ffd2128ffe6a8cf3e871"
    },
    {
      "user_id": "bb5476fd12884539b41d5a88f838d773",
      "links": {
        "self": "http://localhost:5000/v3/credentials/
3d3367228f9c7665266604462ec60029bcd83ad89614021a80b2eb879c572510"
      },
      "blob": "{\"access\": \"181920\", \"secret\": \"secretKey\"}\",
      "project_id": "731fc6f265cd486d900f16e84c5cb594",
      "type": "ec2",
      "id":
      "3d3367228f9c7665266604462ec60029bcd83ad89614021a80b2eb879c572510"
    },
    {
      "user_id": "bb5476fd12884539b41d5a88f838d773",
      "links": {
        "self": "http://localhost:5000/v3/credentials/
6b7d803fc03b85866904b6b79e0a8fa1f4013b584163b4477eed96717eb402c0"
      },
      "blob": "{\"access\": \"f2ba45670b504a518b46e920d760fde2\", \"
\"secret\": \"bf7fff2b3a844730b2db793411756e55\", \"trust_id\": null}\",
      "project_id": "731fc6f265cd486d900f16e84c5cb594",
      "type": "ec2",
      "id":
      "6b7d803fc03b85866904b6b79e0a8fa1f4013b584163b4477eed96717eb402c0"
    },
    {
      "user_id": "2b657f6742ac416697e6821b3b2ee785",
      "links": {
        "self": "http://localhost:5000/v3/credentials/
7d391b869631e5c4836708ea3bb3e0a5cbe0481201b5f0ddd5685ad3b3faa564"
      },
      "blob": "{\"access\": \"a1525da4e7c0438ebf3058372d637b59\", \"
\"secret\": \"c9165d2542b141e8b2a1ff61a5f5487c\", \"trust_id\": null}\",
      "project_id": "2bf9767f9db949ee8364262a28a23062",

```

```

        "type": "ec2",
        "id":
        "7d391b869631e5c4836708ea3bb3e0a5cbe0481201b5f0ddd5685ad3b3faa564"
    },
    {
        "user_id": "bb5476fd12884539b41d5a88f838d773",
        "links": {
            "self": "http://localhost:5000/v3/credentials/
7ef4faa904ae7b8b4ddc7bad15b05ee359dad7d7a9b82861d4ad92fdbbb2eb4e"
        },
        "blob": "{\\"access\\": \\"7d7559359b57419eb5f5f5dcd65ab57d\\", \\"
secret\\": \\"570652bcf8c2483c86eb29e9734eed3c\\", \\"trust_id\\": null}",
        "project_id": "731fc6f265cd486d900f16e84c5cb594",
        "type": "ec2",
        "id":
        "7ef4faa904ae7b8b4ddc7bad15b05ee359dad7d7a9b82861d4ad92fdbbb2eb4e"
    },
    {
        "user_id": "aedb193e9bb8400485f8d8426f7a031f",
        "links": {
            "self": "http://localhost:5000/v3/credentials/
9c1c428d8e0e8338a5e16489ecffff9962f2b00f984ce4c7e9015e4003f478df8"
        },
        "blob": "{\\"access\\": \\"b3a6e5f4427c47e9b202264d91a19e49\\", \\"
secret\\": \\"d9eb470f503f4b46932de38db7a79402\\", \\"trust_id\\": null}",
        "project_id": "a2672ecf9dd34c6980448b25a47e0947",
        "type": "ec2",
        "id":
        "9c1c428d8e0e8338a5e16489ecffff9962f2b00f984ce4c7e9015e4003f478df8"
    },
    {
        "user_id": "c14107e65d5c4a7f8894fc4b3fc209ff",
        "links": {
            "self": "http://localhost:5000/v3/credentials/
e2c35ac2becb0fca3c3c2f035692a4f46a9cbf3b6e86c8a47f5aafe837d78a05"
        },
        "blob": "{\\"access\\": \\"1ed843b1bd4a409f9562400085adbaa4\\", \\"
secret\\": \\"236ab24db1f04ec995fcf618ed4fc0f5\\", \\"trust_id\\": null}",
        "project_id": "6e01855f345f4c59812999b5e459137d",
        "type": "ec2",
        "id":
        "e2c35ac2becb0fca3c3c2f035692a4f46a9cbf3b6e86c8a47f5aafe837d78a05"
    }
},
"links": {
    "self": "http://localhost:5000/v3/credentials",
    "previous": null,
    "next": null
}
}

```


7.2.3. Show credential details

Method	URI	Description
GET	/v3/credentials/{credential_id}	Shows details for a credential.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.2.3.1. Request

This table shows the URI parameters for the show credential details request:

Name	Type	Description
{credential_id}	UUID	The UUID for the credential.

This operation does not accept a request body.

7.2.3.2. Response

Example 7.19. Show credential details: JSON response

```
{
  "credential": {
    "user_id": "bb5476fd12884539b41d5a88f838d773",
    "links": {
      "self": "http://localhost:5000/v3/credentials/207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7"
    },
    "blob": "{\"access\": \"a42a27755ce6442596b049bd7dd8a563\", \"secret\": \"71faf1d40bb24c82b479b1c6fbbd9f0c\", \"trust_id\": null}\",
    \"project_id\": \"6e01855f345f4c59812999b5e459137d\",
    \"type\": \"ec2\",
    \"id\": \"207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7\"
  }
}
```

7.2.4. Update credential

Method	URI	Description
PATCH	/v3/credentials/{credential_id}	Updates a credential.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.2.4.1. Request

This table shows the URI parameters for the update credential request:

Name	Type	Description
{credential_id}	UUID	The UUID for the credential.

Example 7.20. Update credential: JSON request

```
{
  "credential": {
    "blob": "{\"access\":\"181920\\\", \"secrete\":\"secretKey\\\"}\",
    "project_id": "731fc6f265cd486d900f16e84c5cb594",
    "type": "ec2",
    "user_id": "bb5476fd12884539b41d5a88f838d773"
  }
}
```

7.2.4.2. Response

Example 7.21. Update credential: JSON response

```
{
  "credential": {
    "user_id": "bb5476fd12884539b41d5a88f838d773",
    "links": {
      "self": "http://localhost:5000/v3/credentials/207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7"
    },
    "blob": "{\"access\":\"181920\\\", \"secrete\":\"secretKey\\\"}\",
    "project_id": "731fc6f265cd486d900f16e84c5cb594",
    "type": "ec2",
    "id": "207e9b76935efc03804d3dd6ab52d22e9b22a0711e4ada4ff8b76165a07311d7"
  }
}
```

7.2.5. Delete credential

Method	URI	Description
DELETE	/v3/credentials/{credential_id}	Deletes a credential.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.2.5.1. Request

This table shows the URI parameters for the delete credential request:

Name	Type	Description
{credential_id}	UUID	The UUID for the credential.

This operation does not accept a request body.

7.3. Domains (domains, users, groups, roles)

A domain is a collection of users, groups, and projects. Each group and project is owned by exactly one domain.

Each domain defines a namespace where certain API-visible name attributes exist, which affects whether those names must be globally unique or unique within that domain. In the Identity API, the uniqueness of these attributes is as follows:

- *Domain name.* Globally unique across all domains.
- *Role name.* Globally unique across all domains.
- *User name.* Unique within the owning domain.
- *Project name.* Unique within the owning domain.
- *Group name.* Unique within the owning domain.

Method	URI	Description
GET	/v3/domains{?name,enabled}	Lists all domains.
POST	/v3/domains	Creates a domain.
GET	/v3/domains/{domain_id}	Shows details for a domain.
PATCH	/v3/domains/{domain_id}	Updates a domain.
DELETE	/v3/domains/{domain_id}	Deletes a domain.

7.3.1. List domains

Method	URI	Description
GET	/v3/domains{?name,enabled}	Lists all domains.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.3.1.1. Request

This table shows the query parameters for the list domains request:

Name	Type	Description
name	String (Optional)	Filters the response by a domain name.
enabled	String (Optional)	Filters the response by either enabled (true) or disabled (false) domains. Users can authorize against an enabled domain and any of its projects. Users that are owned by an enabled domain can authenticate and receive additional authorization. Users cannot authorize against a disabled domain or any of its projects. Users that are owned by a disabled domain cannot authenticate or receive additional authorization. All tokens that are authorized for a disabled domain or its projects become no longer valid. If you reenable the domain, these tokens are not re-enabled.

This operation does not accept a request body.

7.3.1.2. Response

Example 7.22. List domains: JSON response

```
{
  "domains": [
    {
      "description": "Used for swift functional testing",
      "enabled": true,
      "id": "5a75994a383c449184053ff7270c4e91",
      "links": {
        "self": "http://localhost:5000/v3/domains/5a75994a383c449184053ff7270c4e91"
      },
      "name": "swift_test"
    },
    {
      "description": "Owns users and tenants (i.e. projects) available on Identity API v2.",
      "enabled": true,
      "id": "default",
      "links": {
```

```
        "self": "http://localhost:5000/v3/domains/default"
      },
      "name": "Default"
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "http://localhost:5000/v3/domains"
  }
}
```

7.3.2. Create domain

Method	URI	Description
POST	/v3/domains	Creates a domain.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.3.2.1. Request

Example 7.23. Create domain: JSON request

```
{
  "domain": {
    "description": "Domain description",
    "enabled": true,
    "name": "myDomain"
  }
}
```

7.3.2.2. Response

Example 7.24. Create domain: JSON response

```
{
  "domain": {
    "description": "Domain description",
    "enabled": true,
    "id": "161718",
    "links": {
      "self": "http://identity:35357/v3/domains/161718"
    },
    "name": "myDomain"
  }
}
```

7.3.3. Show domain details

Method	URI	Description
GET	/v3/domains/{domain_id}	Shows details for a domain.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.3.3.1. Request

This table shows the URI parameters for the show domain details request:

Name	Type	Description
{domain_id}	UUID	The domain ID.

This operation does not accept a request body.

7.3.3.2. Response

Example 7.25. Show domain details: JSON response

```
{
  "domain": {
    "description": "Owns users and tenants (i.e. projects) available on
Identity API v2.",
    "enabled": true,
    "id": "default",
    "links": {
      "self": "http://localhost:5000/v3/domains/default"
    },
    "name": "Default"
  }
}
```

7.3.4. Update domain

Method	URI	Description
PATCH	/v3/domains/{domain_id}	Updates a domain.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.3.4.1. Request

This table shows the URI parameters for the update domain request:

Name	Type	Description
{domain_id}	UUID	The domain ID.

Example 7.26. Update domain: JSON request

```
{
  "domain": {
    "description": "Owns users and projects on Identity API v2."
  }
}
```

7.3.4.2. Response

Example 7.27. Update domain: JSON response

```
{
  "domain": {
    "links": {
      "self": "http://localhost:5000/v3/domains/default"
    },
    "enabled": true,
    "description": "Owns users and projects on Identity API v2.",
    "name": "Default",
    "id": "default"
  }
}
```


7.3.5. Delete domain

Method	URI	Description
DELETE	/v3/domains/{domain_id}	Deletes a domain.

To minimize the risk of accidentally deleting a domain, you must first disable the domain by using the update domain method.



Warning

When you delete a domain, this call also deletes all entities owned by it, such as users, groups, and projects, and any credentials and granted roles that relate to those entities.

(Since v3.6) The deletion of a non-leaf domain in a domain hierarchy tree is not allowed and fails with a Bad Request (400) response code.

If you try to delete an enabled domain, this call returns the Forbidden (403) response code.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.3.5.1. Request

This table shows the URI parameters for the delete domain request:

Name	Type	Description
{domain_id}	UUID	The domain ID.

This operation does not accept a request body.

7.4. Domain configuration (domains, config) (since v3.4) (EXPERIMENTAL)

You can manage domain-specific configuration options.

Domain-specific configuration options are structured within their group objects. The API supports only the `identity` and `ldap` groups. These groups override the default configuration settings for the storage of users and groups by the Identity server.

You can create, update, and delete domain-specific configuration options by using the HTTP **PUT**, **PATCH**, and **DELETE** methods. When updating, it is only necessary to include those options that are being updated.

To create an option, use the **PUT** method. The Identity API does not return options that are considered sensitive, although you can create and update these options. The only option currently considered sensitive is the `password` option within the `ldap` group.

The API enables you to include sensitive options as part of non-sensitive options. For example, you can include the password as part of the `url` option.

If you try to create or update configuration options for groups other than the `identity` or `ldap` groups, the `Forbidden (403)` response code is returned.

For information about how to integrate the Identity service with LDAP, see [Integrate Identity with LDAP](#).

Method	URI	Description
GET	<code>/v3/domains/{domain_id}/config</code>	Shows details for a domain configuration.
PATCH	<code>/v3/domains/{domain_id}/config</code>	Updates a domain configuration.
DELETE	<code>/v3/domains/{domain_id}/config</code>	Deletes a domain configuration.
GET	<code>/v3/domains/{domain_id}/config/{group}</code>	Shows details for a domain group configuration.
PATCH	<code>/v3/domains/{domain_id}/config/{group}</code>	Updates a domain group configuration.
DELETE	<code>/v3/domains/{domain_id}/config/{group}</code>	Deletes a domain group configuration.
GET	<code>/v3/domains/{domain_id}/config/{group}/{option}</code>	Shows details for a domain group option configuration.
PATCH	<code>/v3/domains/{domain_id}/config/{group}/{option}</code>	Updates a domain group option configuration.
DELETE	<code>/v3/domains/{domain_id}/config/{group}/{option}</code>	Deletes a domain group option configuration.

7.4.1. Show domain configuration

Method	URI	Description
GET	/v3/domains/{domain_id}/config	Shows details for a domain configuration.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.4.1.1. Request

This table shows the URI parameters for the show domain configuration request:

Name	Type	Description
{domain_id}	UUID	The domain ID.

This operation does not accept a request body.

7.4.1.2. Response

Example 7.28. Show domain configuration: JSON response

```
{
  "config": {
    "identity": {
      "driver": "keystone.identity.backends.ldap.Identity"
    },
    "ldap": {
      "url": "http://myldap/root",
      "user_tree_dn": "ou=Users,dc=root,dc=org"
    }
  }
}
```

7.4.2. Update domain configuration

Method	URI	Description
PATCH	/v3/domains/{domain_id}/config	Updates a domain configuration.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.4.2.1. Request

This table shows the URI parameters for the update domain configuration request:

Name	Type	Description
{domain_id}	UUID	The domain ID.

Example 7.29. Update domain configuration: JSON request

```
{
  "config": {
    "ldap": {
      "url": "http://myldap/my_new_root",
      "user_tree_dn": "ou=Users,dc=my_new_root,dc=org"
    }
  }
}
```

7.4.2.2. Response

Example 7.30. Update domain configuration: JSON response

```
{
  "config": {
    "identity": {
      "driver": "keystone.identity.backends.ldap.Identity"
    },
    "ldap": {
      "url": "http://myldap/my_new_root",
      "user_tree_dn": "ou=Users,dc=my_new_root,dc=org"
    }
  }
}
```

7.4.3. Delete domain configuration

Method	URI	Description
DELETE	/v3/domains/{domain_id}/config	Deletes a domain configuration.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.4.3.1. Request

This table shows the URI parameters for the delete domain configuration request:

Name	Type	Description
{domain_id}	UUID	The domain ID.

This operation does not accept a request body.

7.4.4. Show domain group configuration

Method	URI	Description
GET	/v3/domains/{domain_id}/config/{group}	Shows details for a domain group configuration.

The API supports only the `identity` and `ldap` groups.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.4.4.1. Request

This table shows the URI parameters for the show domain group configuration request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{group}	UUID	The group name, which is <code>ldap</code> or <code>identity</code> .

This operation does not accept a request body.

7.4.4.2. Response

Example 7.31. Show domain group configuration: JSON response

```
{
  "ldap": {
    "url": "http://myldap/root",
    "user_tree_dn": "ou=Users,dc=root,dc=org"
  }
}
```

7.4.5. Update domain group configuration

Method	URI	Description
PATCH	/v3/domains/{domain_id}/config/{group}	Updates a domain group configuration.

The API supports only the `identity` and `ldap` groups. If you try to set configuration options for other groups, this call fails with the `Forbidden` (403) response code.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.4.5.1. Request

This table shows the URI parameters for the update domain group configuration request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{group}	UUID	The group name, which is <code>ldap</code> or <code>identity</code> .

Example 7.32. Update domain group configuration: JSON request

```
{
  "config": {
    "ldap": {
      "url": "http://myldap/my_new_root",
      "user_tree_dn": "ou=Users,dc=my_new_root,dc=org"
    }
  }
}
```

7.4.5.2. Response

Example 7.33. Update domain group configuration: JSON response

```
{
  "config": {
    "identity": {
      "driver": "keystone.identity.backends.ldap.Identity"
    },
    "ldap": {
      "url": "http://myldap/my_new_root",
      "user_tree_dn": "ou=Users,dc=my_new_root,dc=org"
    }
  }
}
```

7.4.6. Delete domain group configuration

Method	URI	Description
DELETE	/v3/domains/{domain_id}/config/{group}	Deletes a domain group configuration.

The API supports only the `identity` and `ldap` groups.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.4.6.1. Request

This table shows the URI parameters for the delete domain group configuration request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{group}	UUID	The group name, which is <code>ldap</code> or <code>identity</code> .

This operation does not accept a request body.

7.4.7. Show domain group option configuration

Method	URI	Description
GET	/v3/domains/{domain_id}/config/{group}/{option}	Shows details for a domain group option configuration.

The API supports only the `identity` and `ldap` groups. For the `ldap` group, a valid value is `url` or `user_tree_dn`. For the `identity` group, a valid value is `driver`.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.4.7.1. Request

This table shows the URI parameters for the show domain group option configuration request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{group}	UUID	The group name, which is <code>ldap</code> or <code>identity</code> .
{option}	UUID	The option name. For the <code>ldap</code> group, a valid value is <code>url</code> or <code>user_tree_dn</code> . For the <code>identity</code> group, a valid value is <code>driver</code> .

This operation does not accept a request body.

7.4.7.2. Response

Example 7.34. Show domain group option configuration: JSON response

```
{
  "url": "http://myldap/root"
}
```

7.4.8. Update domain group option configuration

Method	URI	Description
PATCH	/v3/domains/{domain_id}/config/{group}/{option}	Updates a domain group option configuration.

The API supports only the `identity` and `ldap` groups. For the `ldap` group, a valid value is `url` or `user_tree_dn`. For the `identity` group, a valid value is `driver`.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.4.8.1. Request

This table shows the URI parameters for the update domain group option configuration request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{group}	UUID	The group name, which is <code>ldap</code> or <code>identity</code> .
{option}	UUID	The option name. For the <code>ldap</code> group, a valid value is <code>url</code> or <code>user_tree_dn</code> . For the <code>identity</code> group, a valid value is <code>driver</code> .

Example 7.35. Update domain group option configuration: JSON request

```
{
  "url": "http://myldap/my_other_root"
}
```

7.4.8.2. Response

Example 7.36. Update domain group option configuration: JSON response

```
{
  "config": {
    "identity": {
      "driver": "keystone.identity.backends.ldap.Identity"
    },
    "ldap": {
      "url": "http://myldap/my_other_root",
      "user_tree_dn": "ou=Users,dc=my_new_root,dc=org"
    }
  }
}
```

7.4.9. Delete domain group option configuration

Method	URI	Description
DELETE	/v3/domains/{domain_id}/config/{group}/{option}	Deletes a domain group option configuration.

The API supports only the `identity` and `ldap` groups. For the `ldap` group, a valid value is `url` or `user_tree_dn`. For the `identity` group, a valid value is `driver`.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.4.9.1. Request

This table shows the URI parameters for the delete domain group option configuration request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{group}	UUID	The group name, which is <code>ldap</code> or <code>identity</code> .
{option}	UUID	The option name. For the <code>ldap</code> group, a valid value is <code>url</code> or <code>user_tree_dn</code> . For the <code>identity</code> group, a valid value is <code>driver</code> .

This operation does not accept a request body.

7.5. Groups (groups, users)

A group is a collection of users. Each group is owned by a domain.

When you grant a role to a group, you explicitly associate that group with a project or domain. This action is equivalent to granting the role to each group member on that project and domain.

When you grant a role to a group, that role is automatically granted to any user that you add to the group. When you revoke a role from a group, that role is automatically revoked from any user that you remove from the group. Any token that authenticates that user, project, and domain is revoked.

As with users, a group without any role grants is useless from the perspective an OpenStack service and has no access to resources. However, a group without role grants is permitted as a way of acquiring or loading users and groups from external sources before mapping them to projects and domains.

Method	URI	Description
POST	/v3/groups	Creates a group.
GET	/v3/groups{?domain_id,name}	Lists groups.
GET	/v3/groups/{group_id}	Shows details for a group.

Method	URI	Description
PATCH	/v3/groups/{group_id}	Updates a group.
DELETE	/v3/groups/{group_id}	Deletes a group.
GET	/v3/groups/{group_id}/users{? domain_id,description,name,en- abled}	Lists the users that belong to a group.
PUT	/v3/groups/{group_id}/users/ {user_id}	Adds a user to a group.
HEAD	/v3/groups/{group_id}/users/ {user_id}	Validates that a user belongs to a group.
DELETE	/v3/groups/{group_id}/users/ {user_id}	Removes a user from a group.

7.5.1. Create group

Method	URI	Description
POST	/v3/groups	Creates a group.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.5.1.1. Request

Example 7.37. Create group: JSON request

```
{
  "group": {
    "description": "Contract developers",
    "domain_id": "default",
    "name": "Contract developers"
  }
}
```

7.5.1.2. Response

Example 7.38. Create group: JSON response

```
{
  "group": {
    "domain_id": "default",
    "description": "Contract developers",
    "id": "c0d675eac29945ad9dfd08aa1bb75751",
    "links": {
      "self": "http://localhost:5000/v3/groups/c0d675eac29945ad9dfd08aa1bb75751"
    },
    "name": "Contract developers"
  }
}
```

7.5.2. List groups

Method	URI	Description
GET	/v3/groups{?domain_id,name}	Lists groups.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.5.2.1. Request

This table shows the query parameters for the list groups request:

Name	Type	Description
domain_id	UUID (Optional)	Filters the response by a domain ID.
name	String (Optional)	Filters the response by a group name.

This operation does not accept a request body.

7.5.2.2. Response

Example 7.39. List groups: JSON response

```
{
  "links": {
    "self": "http://localhost:5000/v3/groups",
    "previous": null,
    "next": null
  },
  "groups": [
    {
      "domain_id": "default",
      "description": "non-admin group",
      "id": "96372bbb152f475aa37e9a76a25a029c",
      "links": {
        "self": "http://localhost:5000/v3/groups/96372bbb152f475aa37e9a76a25a029c"
      },
      "name": "nonadmins"
    },
    {
      "domain_id": "default",
      "description": "openstack admin group",
      "id": "9ce0ad4e58a84d7a97b92f7955d10c92",
      "links": {
        "self": "http://localhost:5000/v3/groups/9ce0ad4e58a84d7a97b92f7955d10c92"
      },
      "name": "admins"
    }
  ]
}
```

```
}  
    ]  
}
```

7.5.3. Show group details

Method	URI	Description
GET	/v3/groups/{group_id}	Shows details for a group.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.5.3.1. Request

This table shows the URI parameters for the show group details request:

Name	Type	Description
{group_id}	UUID	The group ID.

This operation does not accept a request body.

7.5.3.2. Response

Example 7.40. Show group details: JSON response

```
{
  "group": {
    "domain_id": "default",
    "description": "Contract developers",
    "id": "c0d675eac29945ad9dfd08aa1bb75751",
    "links": {
      "self": "http://localhost:5000/v3/groups/c0d675eac29945ad9dfd08aa1bb75751"
    },
    "name": "Contract developers"
  }
}
```


7.5.4. Update group

Method	URI	Description
PATCH	/v3/groups/{group_id}	Updates a group.

If the back-end driver does not support this functionality, the call returns the Not Implemented (501) response code.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.5.4.1. Request

This table shows the URI parameters for the update group request:

Name	Type	Description
{group_id}	UUID	The group ID.

Example 7.41. Update group: JSON request

```
{
  "group": {
    "description": "Contract developers 2016",
    "name": "Contract developers 2016"
  }
}
```

7.5.4.2. Response

Example 7.42. Update group: JSON response

```
{
  "group": {
    "domain_id": "default",
    "description": "Contract developers 2016",
    "id": "c0d675eac29945ad9dfd08aa1bb75751",
    "links": {
      "self": "http://localhost:5000/v3/groups/c0d675eac29945ad9dfd08aa1bb75751"
    },
    "name": "Contract developers 2016"
  }
}
```

7.5.5. Delete group

Method	URI	Description
DELETE	/v3/groups/{group_id}	Deletes a group.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.5.5.1. Request

This table shows the URI parameters for the delete group request:

Name	Type	Description
{group_id}	UUID	The group ID.

This operation does not accept a request body.

7.5.6. List users in group

Method	URI	Description
GET	/v3/groups/{group_id}/users{?domain_id,description,name,enabled}	Lists the users that belong to a group.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.5.6.1. Request

This table shows the URI parameters for the list users in group request:

Name	Type	Description
{group_id}	UUID	The group ID.

This operation does not accept a request body.

7.5.6.2. Response

Example 7.43. List users in group: JSON response

```
{
  "users": [
    {
      "name": "admin",
      "links": {
        "self": "http://localhost:5000/v3/users/
fff603a0829d41e48bc0dd0d72ad61ce"
      },
      "domain_id": "default",
      "enabled": true,
      "email": null,
      "id": "fff603a0829d41e48bc0dd0d72ad61ce"
    }
  ],
  "links": {
    "self": "http://localhost:5000/v3/groups/
9ce0ad4e58a84d7a97b92f7955d10c92/users",
    "previous": null,
    "next": null
  }
}
```

7.5.7. Add user to group

Method	URI	Description
PUT	/v3/groups/{group_id}/users/{user_id}	Adds a user to a group.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.5.7.1. Request

This table shows the URI parameters for the add user to group request:

Name	Type	Description
{group_id}	UUID	The group ID.
{user_id}	UUID	The user ID.

This operation does not accept a request body.

7.5.8. Check whether user belongs to group

Method	URI	Description
HEAD	/v3/groups/{group_id}/users/{user_id}	Validates that a user belongs to a group.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.5.8.1. Request

This table shows the URI parameters for the check whether user belongs to group request:

Name	Type	Description
{group_id}	UUID	The group ID.
{user_id}	UUID	The user ID.

This operation does not accept a request body.

7.5.9. Remove user from group

Method	URI	Description
DELETE	/v3/groups/{group_id}/users/{user_id}	Removes a user from a group.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.5.9.1. Request

This table shows the URI parameters for the remove user from group request:

Name	Type	Description
{group_id}	UUID	The group ID.
{user_id}	UUID	The user ID.

This operation does not accept a request body.

7.6. Policies (policies)

A policy is an arbitrarily serialized policy engine rule set to be consumed by a remote service.

You encode policy rule sets into a blob that remote services can consume. To do so, set type to `application/json` and specify policy rules as JSON strings in a blob. For example:

```
{
  "blob": {
    "foobar_user": [
      "role:compute-user"
    ]
  }
}
```

Method	URI	Description
POST	/v3/policies	Creates a policy.
GET	/v3/policies{?type}	Lists policies.
GET	/v3/policies/{policy_id}	Shows details for a policy.
PATCH	/v3/policies/{policy_id}	Updates a policy.
DELETE	/v3/policies/{policy_id}	Deletes a policy.

7.6.1. Create policy

Method	URI	Description
POST	/v3/policies	Creates a policy.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.6.1.1. Request

Example 7.44. Create policy: JSON request

```
{
  "policy": {
    "blob": "{\"foobar_user': 'role:compute-user'}",
    "project_id": "0426ac1e48f642ef9544c2251e07e261",
    "type": "application/json",
    "user_id": "0fffd248c55b443eaac5253b4e9cbf9b5"
  }
}
```

7.6.1.2. Response

Example 7.45. Create policy: JSON response

```
{
  "policy": {
    "user_id": "0fffd248c55b443eaac5253b4e9cbf9b5",
    "links": {
      "self": "http://identity:35357/v3/policies/88f5b83f8f8e41daba4c25eed1a7bbc6"
    },
    "blob": "{\"foobar_user': 'role:compute-user'}",
    "project_id": "0426ac1e48f642ef9544c2251e07e261",
    "type": "application/json",
    "id": "88f5b83f8f8e41daba4c25eed1a7bbc6"
  }
}
```

7.6.2. List policies

Method	URI	Description
GET	/v3/policies{?type}	Lists policies.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.6.2.1. Request

This table shows the query parameters for the list policies request:

Name	Type	Description
type	String (Optional)	Filters the response by a MIME media type for the serialized policy blob. For example, application/json.

This operation does not accept a request body.

7.6.2.2. Response

Example 7.46. List policies: JSON response

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://localhost:5000/v3/policies"
  },
  "policies": [
    {
      "blob": {
        "foobar_user": [
          "role:compute-user"
        ]
      },
      "id": "717273",
      "links": {
        "self": "http://identity:35357/v3/policies/717273"
      },
      "project_id": "456789",
      "type": "application/json",
      "user_id": "616263"
    },
    {
      "blob": {
        "foobar_user": [
          "role:compute-user"
        ]
      },
      "id": "717274",
      "links": {
```



```
        "self": "http://identity:35357/v3/policies/717274"
      },
      "project_id": "456789",
      "type": "application/json",
      "user_id": "616263"
    }
  ]
}
```

7.6.3. Show policy details

Method	URI	Description
GET	/v3/policies/{policy_id}	Shows details for a policy.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.6.3.1. Request

This table shows the URI parameters for the show policy details request:

Name	Type	Description
{policy_id}	UUID	The policy ID.

This operation does not accept a request body.

7.6.3.2. Response

Example 7.47. Show policy details: JSON response

```
{
  "policy": {
    "blob": {
      "foobar_user": [
        "role:compute-user"
      ]
    },
    "id": "717273",
    "links": {
      "self": "http://identity:35357/v3/policies/717273"
    },
    "project_id": "456789",
    "type": "application/json",
    "user_id": "616263"
  }
}
```

7.6.4. Update policy

Method	URI	Description
PATCH	/v3/policies/{policy_id}	Updates a policy.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.6.4.1. Request

This table shows the URI parameters for the update policy request:

Name	Type	Description
{policy_id}	UUID	The policy ID.

Example 7.48. Update policy: JSON request

```
{
  "policy": {
    "blob": {
      "foobar_user": [
        "role:compute-user"
      ]
    },
    "project_id": "456789",
    "type": "application/json",
    "user_id": "616263"
  }
}
```

7.6.4.2. Response

Example 7.49. Update policy: JSON response

```
{
  "policy": {
    "blob": {
      "foobar_user": [
        "role:compute-user"
      ]
    },
    "id": "717273",
    "links": {
      "self": "http://identity:35357/v3/policies/717273"
    },
    "project_id": "456789",
    "type": "application/json",
    "user_id": "616263"
  }
}
```

7.6.5. Delete policy

Method	URI	Description
DELETE	/v3/policies/{policy_id}	Deletes a policy.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.6.5.1. Request

This table shows the URI parameters for the delete policy request:

Name	Type	Description
{policy_id}	UUID	The policy ID.

This operation does not accept a request body.

7.7. Projects (projects, users, groups, roles)

A project is the base unit of ownership. Resources are owned by a specific project. A project is owned by a specific domain.

(Since v3.4) You can create a hierarchy of projects by setting a parent ID when you create projects. All projects in a hierarchy must be owned by the same domain.

(Since v3.6) Optionally, you can create a project that behaves both as a project and a domain. As a domain, the project provides a name space in which you can create users, groups, and other projects. If you create a project that behaves as a domain, you cannot update this project to behave like a regular project.

Method	URI	Description
GET	/v3/projects{?domain_id,parent_id,name,enabled}	Lists projects.
POST	/v3/projects	Creates a project.
GET	/v3/projects/{project_id}	Shows details for a project.
PATCH	/v3/projects/{project_id}	Updates a project.
DELETE	/v3/projects/{project_id}	Deletes a project.
PATCH	/v3/projects/{project_id}/cascade	(Since v3.6) Enables or disables a project and its entire subtree.
DELETE	/v3/projects/{project_id}/cascade	(Since v3.6) Deletes a project and its entire subtree.

7.7.1. List projects

Method	URI	Description
GET	/v3/projects{?domain_id,parent_id,name,enabled}	Lists projects.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.7.1.1. Request

This table shows the query parameters for the list projects request:

Name	Type	Description
domain_id	UUID (Optional)	Filters the response by a domain ID.
parent_id	UUID (Optional)	(Since v3.4) Filters the response by a parent ID.
name	String (Optional)	Filters the response by a project name.
enabled	String (Optional)	Filters the response by either enabled (true) or disabled (false) projects.

This operation does not accept a request body.

7.7.1.2. Response

Example 7.50. List projects: JSON response

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://localhost:5000/v3/projects"
  },
  "projects": [
    {
      "description": null,
      "domain_id": "default",
      "enabled": true,
      "id": "0c4e939acacf4376bdcd1129f1a054ad",
      "links": {
        "self": "http://localhost:5000/v3/projects/0c4e939acacf4376bdcd1129f1a054ad"
      },
      "name": "admin",
      "parent_id": null
    },
    {
      "description": null,
```

```

        "domain_id": "default",
        "enabled": true,
        "id": "0cbd49cbf76d405d9c86562e1d579bd3",
        "links": {
            "self": "http://localhost:5000/v3/projects/
0cbd49cbf76d405d9c86562e1d579bd3"
        },
        "name": "demo",
        "parent_id": null
    },
    {
        "description": null,
        "domain_id": "default",
        "enabled": true,
        "id": "2db68fed84324f29bb73130c6c2094fb",
        "links": {
            "self": "http://localhost:5000/v3/projects/
2db68fed84324f29bb73130c6c2094fb"
        },
        "name": "swifttenanttest2",
        "parent_id": null
    },
    {
        "description": null,
        "domain_id": "default",
        "enabled": true,
        "id": "3d594eb0f04741069dbbb521635b21c7",
        "links": {
            "self": "http://localhost:5000/v3/projects/
3d594eb0f04741069dbbb521635b21c7"
        },
        "name": "service",
        "parent_id": null
    },
    {
        "description": null,
        "domain_id": "default",
        "enabled": true,
        "id": "43ebde53fc314b1c9ea2b8c5dc744927",
        "links": {
            "self": "http://localhost:5000/v3/projects/
43ebde53fc314b1c9ea2b8c5dc744927"
        },
        "name": "swifttenanttest1",
        "parent_id": null
    },
    {
        "description": "",
        "domain_id": "1bc2169ca88e4cdaaba46d4c15390b65",
        "enabled": true,
        "id": "4b1eb781a47440acb8af9850103e537f",
        "links": {
            "self": "http://localhost:5000/v3/projects/
4b1eb781a47440acb8af9850103e537f"
        },
        "name": "swifttenanttest4",
        "parent_id": null
    },
    {
        "description": null,

```

```
        "domain_id": "default",
        "enabled": true,
        "id": "5961c443439d4fcebe42643723755e9d",
        "links": {
            "self": "http://localhost:5000/v3/projects/
5961c443439d4fcebe42643723755e9d"
        },
        "name": "invisible_to_admin",
        "parent_id": null
    },
    {
        "description": null,
        "domain_id": "default",
        "enabled": true,
        "id": "fdb8424c4e4f4c0ba32c52e2de3bd80e",
        "links": {
            "self": "http://localhost:5000/v3/projects/
fdb8424c4e4f4c0ba32c52e2de3bd80e"
        },
        "name": "alt_demo",
        "parent_id": null
    }
]
}
```

7.7.2. Create project

Method	URI	Description
POST	/v3/projects	Creates a project.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.7.2.1. Request

Example 7.51. Create project: JSON request

```
{
  "project": {
    "description": "My new project",
    "domain_id": "default",
    "enabled": true,
    "is_domain": true,
    "name": "myNewProject"
  }
}
```

7.7.2.2. Response

Example 7.52. Create project: JSON response

```
{
  "project": {
    "is_domain": true,
    "description": "My new project",
    "links": {
      "self": "http://localhost:5000/v3/projects/93ebbcc35335488b96ff9cd7d18cbb2e"
    },
    "enabled": true,
    "id": "93ebbcc35335488b96ff9cd7d18cbb2e",
    "parent_id": null,
    "domain_id": "default",
    "name": "myNewProject"
  }
}
```


7.7.3. Show project details

Method	URI	Description
GET	/v3/projects/{project_id}	Shows details for a project.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.7.3.1. Request

This table shows the URI parameters for the show project details request:

Name	Type	Description
{project_id}	UUID	The project ID.

This operation does not accept a request body.

7.7.3.2. Response

Example 7.53. Show project details: JSON response

```
{
  "project": {
    "description": null,
    "domain_id": "default",
    "enabled": true,
    "id": "0c4e939acacf4376bdcd1129f1a054ad",
    "links": {
      "self": "http://localhost:5000/v3/projects/0c4e939acacf4376bdcd1129f1a054ad"
    },
    "name": "admin",
    "parent_id": null
  }
}
```

7.7.4. Update project

Method	URI	Description
PATCH	/v3/projects/{project_id}	Updates a project.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.7.4.1. Request

This table shows the URI parameters for the update project request:

Name	Type	Description
{project_id}	UUID	The project ID.

Example 7.54. Update project: JSON request

```
{
  "project": {
    "description": "My updated project",
    "domain_id": "default",
    "enabled": true,
    "name": "myUpdatedProject"
  }
}
```

7.7.4.2. Response

Example 7.55. Update project: JSON response

```
{
  "project": {
    "is_domain": true,
    "description": "My updated project",
    "links": {
      "self": "http://localhost:5000/v3/projects/93ebbcc35335488b96ff9cd7d18cbb2e"
    },
    "extra": {
      "is_domain": true
    },
    "enabled": true,
    "id": "93ebbcc35335488b96ff9cd7d18cbb2e",
    "parent_id": null,
    "domain_id": "default",
    "name": "myUpdatedProject"
  }
}
```

7.7.5. Delete project

Method	URI	Description
DELETE	/v3/projects/{project_id}	Deletes a project.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.7.5.1. Request

This table shows the URI parameters for the delete project request:

Name	Type	Description
{project_id}	UUID	The project ID.

This operation does not accept a request body.

7.7.6. Enable or disable project and its subtree

Method	URI	Description
PATCH	/v3/projects/{project_id}/cascade	(Since v3.6) Enables or disables a project and its entire subtree.

A project subtree includes all projects beneath the parent project in the hierarchy.

If you include attributes other than the `enabled` attribute, this call fails and returns the **Bad Request (400)** response code.

If you perform this action against a project that acts as a domain (`is_domain` is set to `true`), this call fails and returns the **Forbidden (403)** response code.

Normal response codes: 200

Error response codes: **Bad Request (400)**, **Unauthorized (401)**, **Forbidden (403)**, **Not Found (404)**, **Method Not Allowed (405)**, **conflict (409)**, **Request Entity Too Large (413)**, **Unsupported Media Type (415)**, **Service Unavailable (503)**

7.7.6.1. Request

This table shows the URI parameters for the enable or disable project and its subtree request:

Name	Type	Description
{project_id}	UUID	The project ID.

Example 7.56. Enable or disable project and its subtree: JSON request

```
{
  "project": {
    "enabled": true
  }
}
```

7.7.6.2. Response

Example 7.57. Enable or disable project and its subtree: JSON response

```
{
  "project": {
    "is_domain": true,
    "description": "My updated project",
    "links": {
      "self": "http://localhost:5000/v3/projects/93ebbcc35335488b96ff9cd7d18cbb2e"
    },
    "extra": {
      "is_domain": true
    },
    "enabled": true,
    "id": "93ebbcc35335488b96ff9cd7d18cbb2e",
    "parent_id": null,
  }
}
```

```
        "domain_id": "default",  
        "name": "myUpdatedProject"  
    }  
}
```

7.7.7. Delete project subtree

Method	URI	Description
DELETE	/v3/projects/{project_id}/cascade	(Since v3.6) Deletes a project and its entire subtree.

A project subtree includes all projects beneath the parent project in the hierarchy. You must disable the projects in the subtree before you perform this operation.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.7.7.1. Request

This table shows the URI parameters for the delete project subtree request:

Name	Type	Description
{project_id}	UUID	The project ID.

This operation does not accept a request body.

7.8. Regions (regions) (since v3.2)

A region is a general division of an OpenStack deployment. You can associate zero or more sub-regions with a region to create a tree-like structured hierarchy.

Although a region does not have a geographical connotation, a deployment can use a geographical name for a region, such as us-east.

You can list, create, update, show details for, and delete regions.

Method	URI	Description
GET	/v3/regions{?parent_region_id}	Lists regions.
POST	/v3/regions	Creates a region.
GET	/v3/regions/{region_id}	Shows details for a region, by ID.
PATCH	/v3/regions/{region_id}	Updates a region.
DELETE	/v3/regions/{region_id}	Deletes a region.

7.8.1. List regions

Method	URI	Description
GET	/v3/regions{?parent_region_id}	Lists regions.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.8.1.1. Request

This table shows the query parameters for the list regions request:

Name	Type	Description
parent_region_id	UUID (Optional)	Filters the response by a parent region, by ID.

This operation does not accept a request body.

7.8.1.2. Response

Example 7.58. List regions: JSON response

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://localhost:5000/v3/regions"
  },
  "regions": [
    {
      "description": "",
      "id": "RegionOne",
      "links": {
        "self": "http://localhost:5000/v3/regions/RegionOne"
      },
      "parent_region_id": null
    }
  ]
}
```

7.8.2. Create region

Method	URI	Description
POST	/v3/regions	Creates a region.

When you create the region, you can optionally specify a region ID. If you include characters in the region ID that are not allowed in a URI, you must URL-encode the ID. If you omit an ID, the API assigns an ID to the region.

The following errors might occur:

- Not Found (404). The parent region ID does not exist.
- Conflict (409). The parent region ID would form a circular relationship.
- Conflict (409). The user-defined region ID is not unique to the OpenStack deployment.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.8.2.1. Request

Example 7.59. Create region: JSON request

```
{
  "region": {
    "description": "My subregion",
    "id": "RegionOneSubRegion",
    "parent_region_id": "RegionOne"
  }
}
```

7.8.2.2. Response

Example 7.60. Create region: JSON response

```
{
  "region": {
    "parent_region_id": "RegionOne",
    "id": "RegionOneSubRegion",
    "links": {
      "self": "http://localhost:5000/v3/regions/RegionOneSubRegion"
    },
    "description": "My subregion"
  }
}
```


7.8.3. Show region details

Method	URI	Description
GET	/v3/regions/{region_id}	Shows details for a region, by ID.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.8.3.1. Request

This table shows the URI parameters for the show region details request:

Name	Type	Description
{region_id}	UUID	The region ID.

This operation does not accept a request body.

7.8.3.2. Response

Example 7.61. Show region details: JSON response

```
{
  "region": {
    "description": "My subregion 3",
    "id": "RegionThree",
    "links": {
      "self": "http://localhost:5000/v3/regions/RegionThree"
    },
    "parent_region_id": "RegionOne"
  }
}
```

7.8.4. Update region

Method	URI	Description
PATCH	/v3/regions/{region_id}	Updates a region.

You can update the description or parent region ID for a region. You cannot update the region ID.

The following error might occur:

- Not Found (404). The parent region ID does not exist.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.8.4.1. Request

This table shows the URI parameters for the update region request:

Name	Type	Description
{region_id}	UUID	The region ID.

Example 7.62. Update region: JSON request

```
{
  "region": {
    "description": "My subregion 3"
  }
}
```

7.8.4.2. Response

Example 7.63. Update region: JSON response

```
{
  "region": {
    "parent_region_id": "RegionOne",
    "id": "RegionThree",
    "links": {
      "self": "http://localhost:5000/v3/regions/RegionThree"
    },
    "description": "My subregion 3"
  }
}
```

7.8.5. Delete region

Method	URI	Description
DELETE	/v3/regions/{region_id}	Deletes a region.

The following error might occur:

- **Conflict** (409). The region cannot be deleted because it has child regions.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.8.5.1. Request

This table shows the URI parameters for the delete region request:

Name	Type	Description
{region_id}	UUID	The region ID.

This operation does not accept a request body.

7.9. Roles (roles)

Roles grant a user a set of permitted actions for either a specific project or an entire domain.

You can grant roles to a user on a project, including projects owned by other domains.

You can create, list, and delete roles. You can also list roles assigned to a specified domain, project, or user.

You can list role assignments and, since v3.6, all role assignments within a tree of projects. Use the query parameters to filter the list because the role assignments list can be long. Some typical examples are:

- List role assignments for the specified user:

```
GET /role_assignments?user.id={user_id}
```

- List role assignments for the specified project:

```
GET /role_assignments?scope.project.id={project_id}
```

- List all role assignments for a specified project and its sub- projects:

```
GET /role_assignments?scope.project.id={project_id}?include_subtree=true
```

If you specify `include_subtree=true`, you must also specify the `scope.project.id`. Otherwise, this call returns the Bad Request (400) response code.

Each role assignment entity in the collection contains a link to the assignment that created the entity.

Use the `effective` query parameter to list effective assignments at the user, project, and domain level. This parameter allows for the effects of group membership. The group role assignment entities themselves are not returned in the collection. This represents the effective role assignments that would be included in a scoped token.

In the response, the `links` entity section for entities for group members also contains a URL that enables access to the membership of the group.

You can use the other query parameters with the `effective` parameter, such as:

- Determine what a user can actually do:

```
GET /role_assignments?user.id={user_id}&effective
```

- Get the equivalent set of role assignments that are included in a project-scoped token response:

```
GET /role_assignments?user.id={user_id}&scope.project.id={project_id}&effective
```

Method	URI	Description
POST	/v3/roles	Creates a role.
GET	/v3/roles{?name}	Lists roles.
GET	/v3/roles/{role_id}	Shows details for a role.
PATCH	/v3/roles/{role_id}	Updates a role.
DELETE	/v3/roles/{role_id}	Deletes a role.
GET	/v3/domains/{domain_id}/users/{user_id}/roles	Lists roles for a user on a domain.
PUT	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Grants a role to a user on a domain.
HEAD	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Validates that a user has a role on a domain.
DELETE	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Revokes a role from a user on a domain.
GET	/v3/domains/{domain_id}/groups/{group_id}/roles	Lists roles for a group on a domain.
PUT	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Grants a role to a group on a domain.
HEAD	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Validates that a group has a role on a domain.
DELETE	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Revokes a role from a group on a domain.
GET	/v3/projects/{project_id}/users/{user_id}/roles	Lists roles for a user on a project.
PUT	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Grants a role to a user on a project.

Method	URI	Description
HEAD	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Validates that a user has a role on a project.
DELETE	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Revokes a role from a user on a project.
GET	/v3/projects/{project_id}/groups/{group_id}/roles	Lists roles for a group on a project.
PUT	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Grants a role to a group on a project.
HEAD	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Validates that a group has a role on a project.
DELETE	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Revokes a role from a group on a project.
GET	/v3/role_assignments{?group.id,role.id,scope.domain.id,scope.project.id,user.id,effective,include_subtree}	Lists role assignments.

7.9.1. Create role

Method	URI	Description
POST	/v3/roles	Creates a role.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.9.1.1. Request

Example 7.64. Create role: JSON request

```
{
  "role": {
    "name": "developer"
  }
}
```

7.9.1.2. Response

Example 7.65. Create role: JSON response

```
{
  "role": {
    "id": "1e443fa8cee3482a8a2b6954dd5c8f12",
    "links": {
      "self": "http://localhost:5000/v3/roles/1e443fa8cee3482a8a2b6954dd5c8f12"
    },
    "name": "developer"
  }
}
```

7.9.2. List roles

Method	URI	Description
GET	/v3/roles{?name}	Lists roles.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.9.2.1. Request

This table shows the query parameters for the list roles request:

Name	Type	Description
name	String (Optional)	Filters the response by a role name.

This operation does not accept a request body.

7.9.2.2. Response

Example 7.66. List roles: JSON response

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://localhost:5000/v3/roles"
  },
  "roles": [
    {
      "id": "5318e65d75574c17bf5339d3df33a5a3",
      "links": {
        "self": "http://localhost:5000/v3/roles/5318e65d75574c17bf5339d3df33a5a3"
      },
      "name": "admin"
    },
    {
      "id": "642bcfc75c384fd181adf34d9b2df897",
      "links": {
        "self": "http://localhost:5000/v3/roles/642bcfc75c384fd181adf34d9b2df897"
      },
      "name": "anotherrole"
    },
    {
      "id": "779a76d74f544224a7ef8762ca0de627",
      "links": {
        "self": "http://localhost:5000/v3/roles/779a76d74f544224a7ef8762ca0de627"
      }
    }
  ]
}
```

```
        "name": "Member"
      },
      {
        "id": "9fe2ff9ee4384b1894a90878d3e92bab",
        "links": {
          "self": "http://localhost:5000/v3/roles/9fe2ff9ee4384b1894a90878d3e92bab"
        },
        "name": "_member_"
      },
      {
        "id": "ba2dfba61c934ee89e3110de36273229",
        "links": {
          "self": "http://localhost:5000/v3/roles/ba2dfba61c934ee89e3110de36273229"
        },
        "name": "ResellerAdmin"
      },
      {
        "id": "f127b97616f24d3ebceb7be840210adc",
        "links": {
          "self": "http://localhost:5000/v3/roles/f127b97616f24d3ebceb7be840210adc"
        },
        "name": "service"
      }
    ]
  }
}
```


7.9.3. Show role details

Method	URI	Description
GET	/v3/roles/{role_id}	Shows details for a role.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.9.3.1. Request

This table shows the URI parameters for the show role details request:

Name	Type	Description
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.3.2. Response

Example 7.67. Show role details: JSON response

```
{
  "role": {
    "id": "1e443fa8cee3482a8a2b6954dd5c8f12",
    "links": {
      "self": "http://localhost:5000/v3/roles/1e443fa8cee3482a8a2b6954dd5c8f12"
    },
    "name": "Developer"
  }
}
```

7.9.4. Update role

Method	URI	Description
PATCH	/v3/roles/{role_id}	Updates a role.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.9.4.1. Request

This table shows the URI parameters for the update role request:

Name	Type	Description
{role_id}	UUID	The role ID.

Example 7.68. Update role: JSON request

```
{
  "role": {
    "name": "Developer"
  }
}
```

7.9.4.2. Response

Example 7.69. Update role: JSON response

```
{
  "role": {
    "id": "1e443fa8cee3482a8a2b6954dd5c8f12",
    "links": {
      "self": "http://localhost:5000/v3/roles/1e443fa8cee3482a8a2b6954dd5c8f12"
    },
    "name": "Developer"
  }
}
```

7.9.5. Delete role

Method	URI	Description
DELETE	/v3/roles/{role_id}	Deletes a role.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.9.5.1. Request

This table shows the URI parameters for the delete role request:

Name	Type	Description
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.6. List roles for user on domain

Method	URI	Description
GET	/v3/domains/{domain_id}/users/{user_id}/roles	Lists roles for a user on a domain.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.9.6.1. Request

This table shows the URI parameters for the list roles for user on domain request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{user_id}	UUID	The user ID.

This operation does not accept a request body.

7.9.6.2. Response

Example 7.70. List roles for user on domain: JSON response

```
{
  "roles": [
    {
      "id": "123456",
      "links": {
        "self": "http://identity:35357/v3/roles/123456"
      },
      "name": "admin"
    },
    {
      "id": "123457",
      "links": {
        "self": "http://identity:35357/v3/roles/123457"
      },
      "name": "manager"
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/domains/161718/users/313233/roles",
    "previous": null,
    "next": null
  }
}
```

7.9.7. Grant role to user on domain

Method	URI	Description
PUT	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Grants a role to a user on a domain.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.9.7.1. Request

This table shows the URI parameters for the grant role to user on domain request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{user_id}	UUID	The user ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.8. Check whether user has role on domain

Method	URI	Description
HEAD	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Validates that a user has a role on a domain.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.9.8.1. Request

This table shows the URI parameters for the check whether user has role on domain request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{user_id}	UUID	The user ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.9. Revoke role from user on domain

Method	URI	Description
DELETE	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Revokes a role from a user on a domain.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.9.9.1. Request

This table shows the URI parameters for the revoke role from user on domain request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{user_id}	UUID	The user ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.10. List roles for group on domain

Method	URI	Description
GET	/v3/domains/{domain_id}/groups/{group_id}/roles	Lists roles for a group on a domain.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.9.10.1. Request

This table shows the URI parameters for the list roles for group on domain request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{group_id}	UUID	The group ID.

This operation does not accept a request body.

7.9.10.2. Response

Example 7.71. List roles for group on domain: JSON response

```
{
  "roles": [
    {
      "id": "123456",
      "links": {
        "self": "http://identity:35357/v3/roles/123456"
      },
      "name": "admin"
    },
    {
      "id": "123457",
      "links": {
        "self": "http://identity:35357/v3/roles/123457"
      },
      "name": "manager"
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/domains/161718/groups/101112/roles",
    "previous": null,
    "next": null
  }
}
```


7.9.11. Grant role to group on domain

Method	URI	Description
PUT	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Grants a role to a group on a domain.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.9.11.1. Request

This table shows the URI parameters for the grant role to group on domain request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{group_id}	UUID	The group ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.12. Check whether group has role on domain

Method	URI	Description
HEAD	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Validates that a group has a role on a domain.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.9.12.1. Request

This table shows the URI parameters for the check whether group has role on domain request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{group_id}	UUID	The group ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.13. Revoke role from group on domain

Method	URI	Description
DELETE	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Revokes a role from a group on a domain.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.9.13.1. Request

This table shows the URI parameters for the revoke role from group on domain request:

Name	Type	Description
{domain_id}	UUID	The domain ID.
{group_id}	UUID	The group ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.14. List roles for user on project

Method	URI	Description
GET	/v3/projects/{project_id}/users/{user_id}/roles	Lists roles for a user on a project.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.9.14.1. Request

This table shows the URI parameters for the list roles for user on project request:

Name	Type	Description
{project_id}	UUID	The project ID.
{user_id}	UUID	The user ID.

This operation does not accept a request body.

7.9.14.2. Response

Example 7.72. List roles for user on project: JSON response

```
{
  "links": {
    "self": "http://localhost:5000/v3/projects/
9e5a15e2c0dd42aab0990a463e839ac1/users/b964a9e51c0046a4a84d3f83a135a97c/
roles",
    "previous": null,
    "next": null
  },
  "roles": [
    {
      "id": "3b5347fa7a144008ba57c0acea469cc3",
      "links": {
        "self": "http://localhost:5000/v3/roles/
3b5347fa7a144008ba57c0acea469cc3"
      },
      "name": "admin"
    }
  ]
}
```

7.9.15. Grant role to user on project

Method	URI	Description
PUT	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Grants a role to a user on a project.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.9.15.1. Request

This table shows the URI parameters for the grant role to user on project request:

Name	Type	Description
{project_id}	UUID	The project ID.
{user_id}	UUID	The user ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.16. Check whether user has role on project

Method	URI	Description
HEAD	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Validates that a user has a role on a project.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.9.16.1. Request

This table shows the URI parameters for the check whether user has role on project request:

Name	Type	Description
{project_id}	UUID	The project ID.
{user_id}	UUID	The user ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.17. Revoke role from user on project

Method	URI	Description
DELETE	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Revokes a role from a user on a project.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.9.17.1. Request

This table shows the URI parameters for the revoke role from user on project request:

Name	Type	Description
{project_id}	UUID	The project ID.
{user_id}	UUID	The user ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.18. List roles for group on project

Method	URI	Description
GET	/v3/projects/{project_id}/groups/{group_id}/roles	Lists roles for a group on a project.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.9.18.1. Request

This table shows the URI parameters for the list roles for group on project request:

Name	Type	Description
{project_id}	UUID	The project ID.
{group_id}	UUID	The group ID.

This operation does not accept a request body.

7.9.18.2. Response

Example 7.73. List roles for group on project: JSON response

```
{
  "roles": [
    {
      "id": "123456",
      "links": {
        "self": "http://identity:35357/v3/roles/123456"
      },
      "name": "admin"
    },
    {
      "id": "123457",
      "links": {
        "self": "http://identity:35357/v3/roles/123457"
      },
      "name": "manager"
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/projects/456789/groups/101112/roles",
    "previous": null,
    "next": null
  }
}
```


7.9.19. Grant role to group on project

Method	URI	Description
PUT	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Grants a role to a group on a project.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.9.19.1. Request

This table shows the URI parameters for the grant role to group on project request:

Name	Type	Description
{project_id}	UUID	The project ID.
{group_id}	UUID	The group ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.20. Check whether group has role on project

Method	URI	Description
HEAD	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Validates that a group has a role on a project.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.9.20.1. Request

This table shows the URI parameters for the check whether group has role on project request:

Name	Type	Description
{project_id}	UUID	The project ID.
{group_id}	UUID	The group ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.21. Revoke role from group on project

Method	URI	Description
DELETE	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Revokes a role from a group on a project.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.9.21.1. Request

This table shows the URI parameters for the revoke role from group on project request:

Name	Type	Description
{project_id}	UUID	The project ID.
{group_id}	UUID	The group ID.
{role_id}	UUID	The role ID.

This operation does not accept a request body.

7.9.22. List role assignments

Method	URI	Description
GET	/v3/role_assignments{?group.id, role.id, scope.domain.id, scope.project.id, user.id, effective, include_subtree}	Lists role assignments.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.9.22.1. Request

This operation does not accept a request body.

7.9.22.2. Response

Example 7.74. List role assignments: JSON response

```
GET /role_assignments?user.id={user_id}&scope.project.id={project_id}&effective
```

Example 7.75. List role assignments: JSON response

```
{
  "role_assignments": [
    {
      "links": {
        "assignment": "http://identity:35357/v3/domains/161718/users/313233/roles/123456"
      },
      "role": {
        "id": "123456"
      },
      "scope": {
        "domain": {
          "id": "161718"
        }
      },
      "user": {
        "id": "313233"
      }
    },
    {
      "group": {
        "id": "101112"
      },
      "links": {
        "assignment": "http://identity:35357/v3/projects/456789/groups/101112/roles/123456"
      },
      "role": {
```

```

        "id": "123456"
      },
      "scope": {
        "project": {
          "id": "456789"
        }
      }
    },
    "links": {
      "self": "http://identity:35357/v3/role_assignments",
      "previous": null,
      "next": null
    }
  }
}

```

7.10. Service catalog and endpoints (services, endpoints)

A service is an OpenStack web service that you can access through a URL, or endpoint.

A service catalog lists the services that are available to the caller based upon the current authorization.

You can create, list, show details for, update, and delete services. When you create or update a service, you can enable the service, which causes it and its endpoints to appear in the service catalog.

You can create, list, show details for, update, and delete endpoints.

Method	URI	Description
GET	/v3/services{?type}	Lists all services.
POST	/v3/services	Creates a service.
GET	/v3/services/{service_id}	Shows details for a service.
PATCH	/v3/services/{service_id}	Updates a service.
DELETE	/v3/services/{service_id}	Deletes a service.
GET	/v3/endpoints{?interface, service_id}	Lists all available endpoints.
POST	/v3/endpoints	Creates an endpoint.
GET	/v3/endpoints/{endpoint_id}	Shows details for an endpoint.
PATCH	/v3/endpoints/{endpoint_id}	Updates an endpoint.
DELETE	/v3/endpoints/{endpoint_id}	Deletes an endpoint.

7.10.1. List services

Method	URI	Description
GET	/v3/services{?type}	Lists all services.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.10.1.1. Request

This table shows the header parameters for the list services request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the query parameters for the list services request:

Name	Type	Description
type	String (Optional)	Filters the response by a service type. A valid value is compute, ec2, identity, image, network, or volume.

This operation does not accept a request body.

7.10.1.2. Response

Example 7.76. List services: JSON response

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://localhost:5000/v3/services"
  },
  "services": [
    {
      "description": "Nova Compute Service",
      "enabled": true,
      "id": "1999c3a858c7408fb586817620695098",
      "links": {
        "self": "http://localhost:5000/v3/services/1999c3a858c7408fb586817620695098"
      },
      "name": "nova",
      "type": "compute"
    },
    {
      "description": "Cinder Volume Service V2",
      "enabled": true,
      "id": "39216610e75547f1883037e11976fc0f",
      "links": {
```

```

        "self": "http://localhost:5000/v3/services/
39216610e75547f1883037e11976fc0f"
    },
    "name": "cinderv2",
    "type": "volumev2"
},
{
    "description": "Neutron Service",
    "enabled": true,
    "id": "4fe41a27de3341af9100123f765eac0d",
    "links": {
        "self": "http://localhost:5000/v3/services/
4fe41a27de3341af9100123f765eac0d"
    },
    "name": "neutron",
    "type": "network"
},
{
    "description": "EC2 Compatibility Layer",
    "enabled": true,
    "id": "61d3d05bdd1449f18923c83f52a4d762",
    "links": {
        "self": "http://localhost:5000/v3/services/
61d3d05bdd1449f18923c83f52a4d762"
    },
    "name": "ec2",
    "type": "ec2"
},
{
    "description": "Glance Image Service",
    "enabled": true,
    "id": "69afa3d57d1948ea988beeb252bbaa5d",
    "links": {
        "self": "http://localhost:5000/v3/services/
69afa3d57d1948ea988beeb252bbaa5d"
    },
    "name": "glance",
    "type": "image"
},
{
    "description": "Nova Compute Service V2.1",
    "enabled": true,
    "id": "79b691ee7be649d9bf8613efc0960206",
    "links": {
        "self": "http://localhost:5000/v3/services/
79b691ee7be649d9bf8613efc0960206"
    },
    "name": "novav21",
    "type": "computeV21"
},
{
    "description": "Swift Service",
    "enabled": true,
    "id": "92419b70ebe64c6c873bd20b14360e6b",
    "links": {
        "self": "http://localhost:5000/v3/services/
92419b70ebe64c6c873bd20b14360e6b"
    },
    "name": "swift",
    "type": "object-store"
}

```

```
    },
    {
      "description": "Keystone Identity Service",
      "enabled": true,
      "id": "b8f8454fc07b46b781204d2a436f9d1c",
      "links": {
        "self": "http://localhost:5000/v3/services/
b8f8454fc07b46b781204d2a436f9d1c"
      },
      "name": "keystone",
      "type": "identity"
    },
    {
      "description": "Cinder Volume Service",
      "enabled": true,
      "id": "cdda3bea0742407f95e70f4758f46558",
      "links": {
        "self": "http://localhost:5000/v3/services/
cdda3bea0742407f95e70f4758f46558"
      },
      "name": "cinder",
      "type": "volume"
    }
  ]
}
```


7.10.2. Create service

Method	URI	Description
POST	/v3/services	Creates a service.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.10.2.1. Request

This table shows the header parameters for the create service request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

Example 7.77. Create service: JSON request

```
{
  "service": {
    "type": "compute",
    "name": "compute2",
    "description": "Compute service 2"
  }
}
```

7.10.2.2. Response

Example 7.78. Create service: JSON response

```
{
  "service": {
    "name": "compute2",
    "links": {
      "self": "http://localhost:5000/v3/services/3f552eb79c48436db2868e948d8cf330"
    },
    "enabled": true,
    "type": "compute",
    "id": "3f552eb79c48436db2868e948d8cf330",
    "description": "Compute service 2"
  }
}
```

7.10.3. Show service details

Method	URI	Description
GET	/v3/services/{service_id}	Shows details for a service.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.10.3.1. Request

This table shows the header parameters for the show service details request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the show service details request:

Name	Type	Description
{service_id}	UUID	The service ID.

This operation does not accept a request body.

7.10.3.2. Response

Example 7.79. Show service details: JSON response

```
{
  "service": {
    "description": "Keystone Identity Service",
    "enabled": true,
    "id": "686766",
    "links": {
      "self": "http://identity:5000/v3/services/686766"
    },
    "name": "keystone",
    "type": "identity"
  }
}
```

7.10.4. Update service

Method	URI	Description
PATCH	/v3/services/{service_id}	Updates a service.

The request body is the same as the create service request body, except that you include only those attributes that you want to update.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.10.4.1. Request

This table shows the header parameters for the update service request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the update service request:

Name	Type	Description
{service_id}	UUID	The service ID.

Example 7.80. Update service: JSON request

```
{
  "service": {
    "description": "Block Storage Service V2"
  }
}
```

7.10.4.2. Response

Example 7.81. Update service: JSON response

```
{
  "service": {
    "name": "cinderv2",
    "links": {
      "self": "http://localhost:5000/v3/services/5789da9864004dd088fce14c1c626a4b"
    },
    "enabled": true,
    "type": "volumev2",
    "id": "5789da9864004dd088fce14c1c626a4b",
    "description": "Block Storage Service V2"
  }
}
```

7.10.5. Delete service

Method	URI	Description
DELETE	/v3/services/{service_id}	Deletes a service.



Warning

If you try to delete a service that still has associated endpoints, this call either deletes all associated endpoints or fails until all endpoints are deleted.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.10.5.1. Request

This table shows the header parameters for the delete service request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the delete service request:

Name	Type	Description
{service_id}	UUID	The service ID.

This operation does not accept a request body.

7.10.6. List endpoints

Method	URI	Description
GET	/v3/endpoints{?interface, service_id}	Lists all available endpoints.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.10.6.1. Request

This table shows the query parameters for the list endpoints request:

Name	Type	Description
interface	String (Optional)	Filters the response by an interface.
service_id	UUID (Optional)	Filters the response by a service ID.

This operation does not accept a request body.

7.10.6.2. Response

Example 7.82. List endpoints: JSON response

```
{
  "endpoints": [
    {
      "enabled": true,
      "id": "0649c5be323f4792afbc1efdd480847d",
      "interface": "internal",
      "links": {
        "self": "http://localhost:5000/v3/endpoints/0649c5be323f4792afbc1efdd480847d"
      },
      "region": "RegionOne",
      "region_id": "RegionOne",
      "service_id": "ef6b15e425814dc69d830361baae0e33",
      "url": "http://23.253.211.234:8080/v1/AUTH_$(tenant_id)s"
    },
    {
      "enabled": true,
      "id": "06b85ed2aa57413ca0b1813daed329a9",
      "interface": "internal",
      "links": {
        "self": "http://localhost:5000/v3/endpoints/06b85ed2aa57413ca0b1813daed329a9"
      },
      "region": "RegionOne",
      "region_id": "RegionOne",
      "service_id": "98cfd5347fb84601b2f88f3afd8dddd4",
      "url": "http://23.253.211.234:8776/v1/$(tenant_id)s"
    }
  ]
}
```

```

    },
    {
      "enabled": true,
      "id": "070102f162e04f91a52c7887d0604163",
      "interface": "admin",
      "links": {
        "self": "http://localhost:5000/v3/endpoints/070102f162e04f91a52c7887d0604163"
      },
      "region": "RegionOne",
      "region_id": "RegionOne",
      "service_id": "312f401c14d143d8b3e3f4daf0418add",
      "url": "http://23.253.211.234:8774/v2.1/$(tenant_id)s"
    },
    {
      "enabled": true,
      "id": "0fd73b621e424cc0a172853264519cbc",
      "interface": "admin",
      "links": {
        "self": "http://localhost:5000/v3/endpoints/0fd73b621e424cc0a172853264519cbc"
      },
      "region": "RegionOne",
      "region_id": "RegionOne",
      "service_id": "17a877162c8e405b81d563d95ec4e3f8",
      "url": "http://23.253.211.234:8776/v2/$(tenant_id)s"
    },
    {
      "enabled": true,
      "id": "1899667a3b1544ccb355fd4c4184d7d7",
      "interface": "public",
      "links": {
        "self": "http://localhost:5000/v3/endpoints/1899667a3b1544ccb355fd4c4184d7d7"
      },
      "region": "RegionOne",
      "region_id": "RegionOne",
      "service_id": "9b67aed49e0d4c2fb46ca9476a3b9243",
      "url": "http://23.253.211.234:9292"
    },
    {
      "enabled": true,
      "id": "3b3611ea2e554ee7b85e7f2213b02c33",
      "interface": "admin",
      "links": {
        "self": "http://localhost:5000/v3/endpoints/3b3611ea2e554ee7b85e7f2213b02c33"
      },
      "region": "RegionOne",
      "region_id": "RegionOne",
      "service_id": "2a662f90700b4478929d4b24cc6a320b",
      "url": "http://23.253.211.234:9696/"
    },
    {
      "enabled": true,
      "id": "3ea2b420306f48c6bf0cf51c2fefe03",
      "interface": "internal",
      "links": {
        "self": "http://localhost:5000/v3/endpoints/3ea2b420306f48c6bf0cf51c2fefe03"
      }
    }
  ]
}

```

```

        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "736fb9bb21ef498287db9abcc55b20d9",
        "url": "http://23.253.211.234:8774/v2/$(tenant_id)s"
    },
    {
        "enabled": true,
        "id": "41b122182f574a44b0e246aff6ca29c5",
        "interface": "admin",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/
41b122182f574a44b0e246aff6ca29c5"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "9b67aed49e0d4c2fb46ca9476a3b9243",
        "url": "http://23.253.211.234:9292"
    },
    {
        "enabled": true,
        "id": "44a736dd5eeb4347acec66b5f11c8f80",
        "interface": "internal",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/
44a736dd5eeb4347acec66b5f11c8f80"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "2a662f90700b4478929d4b24cc6a320b",
        "url": "http://23.253.211.234:9696/"
    },
    {
        "enabled": true,
        "id": "499e8f6718ef466ba3fb315fa8f9e0b8",
        "interface": "internal",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/
499e8f6718ef466ba3fb315fa8f9e0b8"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "312f401c14d143d8b3e3f4daf0418add",
        "url": "http://23.253.211.234:8774/v2.1/$(tenant_id)s"
    },
    {
        "enabled": true,
        "id": "545b1e9f126248428c5cdbec7420c353",
        "interface": "public",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/
545b1e9f126248428c5cdbec7420c353"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "736fb9bb21ef498287db9abcc55b20d9",
        "url": "http://23.253.211.234:8774/v2/$(tenant_id)s"
    },
    {
        "enabled": true,

```

```

        "id": "629dc5a64e954ad09a45e87bc48299ba",
        "interface": "public",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/
629dc5a64e954ad09a45e87bc48299ba"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "2a662f90700b4478929d4b24cc6a320b",
        "url": "http://23.253.211.234:9696/"
    },
    {
        "enabled": true,
        "id": "642a329a660544fdaab2420c0da7d49b",
        "interface": "public",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/
642a329a660544fdaab2420c0da7d49b"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "ef6b15e425814dc69d830361baae0e33",
        "url": "http://23.253.211.234:8080/v1/AUTH_$(tenant_id)s"
    },
    {
        "enabled": true,
        "id": "72f8fc8536e44a19bc3388218efcc741",
        "interface": "internal",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/
72f8fc8536e44a19bc3388218efcc741"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "9b67aed49e0d4c2fb46ca9476a3b9243",
        "url": "http://23.253.211.234:9292"
    },
    {
        "enabled": true,
        "id": "74121e71962e4947ac622c41706f0ee7",
        "interface": "public",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/
74121e71962e4947ac622c41706f0ee7"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "17a877162c8e405b81d563d95ec4e3f8",
        "url": "http://23.253.211.234:8776/v2/$(tenant_id)s"
    },
    {
        "enabled": true,
        "id": "7431a4f971dc4abb8d0e387434a06817",
        "interface": "admin",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/
7431a4f971dc4abb8d0e387434a06817"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",

```



```

        "service_id": "9242e05f0c23467bbd1cf1f7a6e5e596",
        "url": "http://23.253.211.234:8773/"
    },
    {
        "enabled": true,
        "id": "7cffc75a14ca4334b458e475750bd84f",
        "interface": "public",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/7cffc75a14ca4334b458e475750bd84f"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "efeb249cbcd3412496bc4b194ea058da",
        "url": "http://23.253.211.234:5000/v2.0"
    },
    {
        "enabled": true,
        "id": "a422a6fa163b4a6ba8309e067ce3750b",
        "interface": "public",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/a422a6fa163b4a6ba8309e067ce3750b"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "312f401c14d143d8b3e3f4daf0418add",
        "url": "http://23.253.211.234:8774/v2.1/$(tenant_id)s"
    },
    {
        "enabled": true,
        "id": "ac6a74efe9944afdb129d4df70cde0ec",
        "interface": "public",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/ac6a74efe9944afdb129d4df70cde0ec"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "9242e05f0c23467bbd1cf1f7a6e5e596",
        "url": "http://23.253.211.234:8773/"
    },
    {
        "enabled": true,
        "id": "adf43d7ff0d14d0fa1e8a5187f40e1af",
        "interface": "internal",
        "links": {
            "self": "http://localhost:5000/v3/endpoints/adf43d7ff0d14d0fa1e8a5187f40e1af"
        },
        "region": "RegionOne",
        "region_id": "RegionOne",
        "service_id": "efeb249cbcd3412496bc4b194ea058da",
        "url": "http://23.253.211.234:5000/v2.0"
    },
    {
        "enabled": true,
        "id": "b18be64a118244d39217db72534f8b33",
        "interface": "admin",
        "links": {

```

```

        "self": "http://localhost:5000/v3/endpoints/
b18be64a118244d39217db72534f8b33"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "736fb9bb21ef498287db9abcc55b20d9",
    "url": "http://23.253.211.234:8774/v2/$(tenant_id)s"
},
{
    "enabled": true,
    "id": "c828983c9c214d819674649aa693cdff",
    "interface": "public",
    "links": {
        "self": "http://localhost:5000/v3/endpoints/
c828983c9c214d819674649aa693cdff"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "98cfd5347fb84601b2f88f3afd8dddd4",
    "url": "http://23.253.211.234:8776/v1/$(tenant_id)s"
},
{
    "enabled": true,
    "id": "d062ebdb244f447498768fc0ced32e2d",
    "interface": "admin",
    "links": {
        "self": "http://localhost:5000/v3/endpoints/
d062ebdb244f447498768fc0ced32e2d"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "98cfd5347fb84601b2f88f3afd8dddd4",
    "url": "http://23.253.211.234:8776/v1/$(tenant_id)s"
},
{
    "enabled": true,
    "id": "d281219ec0df4cf2b7c681463d5dcf51",
    "interface": "internal",
    "links": {
        "self": "http://localhost:5000/v3/endpoints/
d281219ec0df4cf2b7c681463d5dcf51"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "17a877162c8e405b81d563d95ec4e3f8",
    "url": "http://23.253.211.234:8776/v2/$(tenant_id)s"
},
{
    "enabled": true,
    "id": "d8e0824a17404431b5d978a87ac1bede",
    "interface": "admin",
    "links": {
        "self": "http://localhost:5000/v3/endpoints/
d8e0824a17404431b5d978a87ac1bede"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "efeb249cbcd3412496bc4b194ea058da",
    "url": "http://23.253.211.234:35357/v2.0"
},

```

```
{
  "enabled": true,
  "id": "d9b54bdc063046828ac3c6487bea8047",
  "interface": "internal",
  "links": {
    "self": "http://localhost:5000/v3/endpoints/
d9b54bdc063046828ac3c6487bea8047"
  },
  "region": "RegionOne",
  "region_id": "RegionOne",
  "service_id": "9242e05f0c23467bbd1cf1f7a6e5e596",
  "url": "http://23.253.211.234:8773/"
},
{
  "enabled": true,
  "id": "ea74f9771dec475eabfc2cdff5364413",
  "interface": "admin",
  "links": {
    "self": "http://localhost:5000/v3/endpoints/
ea74f9771dec475eabfc2cdff5364413"
  },
  "region": "RegionOne",
  "region_id": "RegionOne",
  "service_id": "ef6b15e425814dc69d830361baae0e33",
  "url": "http://23.253.211.234:8080"
}
],
"links": {
  "next": null,
  "previous": null,
  "self": "http://localhost:5000/v3/endpoints"
}
}
```

7.10.7. Create endpoint

Method	URI	Description
POST	/v3/endpoints	Creates an endpoint.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.10.7.1. Request

Example 7.83. Create endpoint: JSON request

```
{
  "endpoint": {
    "interface": "public",
    "name": "ec2-2",
    "region_id": "RegionOne",
    "url": "http://identity:35357/v3/endpoints/828384",
    "service_id": "9242e05f0c23467bbd1cf1f7a6e5e596"
  }
}
```

7.10.7.2. Response

Example 7.84. Create endpoint: JSON response

```
{
  "endpoint": {
    "region_id": "RegionOne",
    "links": {
      "self": "http://localhost:5000/v3/endpoints/e9b475a8742d4ff1a81b353c5a37e138"
    },
    "url": "http://identity:35357/v3/endpoints/828384",
    "region": "RegionOne",
    "enabled": true,
    "interface": "public",
    "service_id": "9242e05f0c23467bbd1cf1f7a6e5e596",
    "id": "e9b475a8742d4ff1a81b353c5a37e138",
    "name": "ec2-2"
  }
}
```

7.10.8. Show endpoint details

Method	URI	Description
GET	/v3/endpoints/{endpoint_id}	Shows details for an endpoint.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.10.8.1. Request

This table shows the URI parameters for the show endpoint details request:

Name	Type	Description
{endpoint_id}	UUID	The endpoint ID.

This operation does not accept a request body.

7.10.8.2. Response

Example 7.85. Show endpoint details: JSON response

```
{
  "endpoint": {
    "enabled": true,
    "id": "01c3d5b92f7841ac83fb4b26173c12c7",
    "interface": "admin",
    "links": {
      "self": "http://localhost:5000/v3/endpoints/01c3d5b92f7841ac83fb4b26173c12c7"
    },
    "region": "RegionOne",
    "region_id": "RegionOne",
    "service_id": "3b2d6ad7e02c4cde8498a547601f1b8f",
    "url": "http://23.253.211.234:9696/"
  }
}
```

7.10.9. Update endpoint

Method	URI	Description
PATCH	/v3/endpoints/{endpoint_id}	Updates an endpoint.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.10.9.1. Request

This table shows the URI parameters for the update endpoint request:

Name	Type	Description
{endpoint_id}	UUID	The endpoint ID.

Example 7.86. Update endpoint: JSON request

```
{
  "endpoint": {
    "interface": "public",
    "name": "Name",
    "region_id": "north",
    "url": "http://identity:35357/v3/endpoints/828384",
    "service_id": "345678"
  }
}
```

7.10.9.2. Response

Example 7.87. Update endpoint: JSON response

```
{
  "endpoint": {
    "id": "828384",
    "interface": "internal",
    "links": {
      "self": "http://identity:35357/v3/endpoints/828384"
    },
    "name": "the internal volume endpoint",
    "region_id": "north",
    "service_id": "686766",
    "url": "http://identity:35357/v3/endpoints/828384"
  }
}
```

7.10.10. Delete endpoint

Method	URI	Description
DELETE	/v3/endpoints/{endpoint_id}	Deletes an endpoint.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.10.10.1. Request

This table shows the URI parameters for the delete endpoint request:

Name	Type	Description
{endpoint_id}	UUID	The endpoint ID.

This operation does not accept a request body.

7.11. Users (users, groups, projects, roles)

A user is an individual API consumer that is owned by a domain. A role explicitly associates a user with projects or domains. A user with no assigned roles has no access to OpenStack resources.

You can list, create, show details for, update, delete, and change the password for users.

You can also list groups, projects, and roles for a specified user. To list user roles, see [Roles](#).

Method	URI	Description
GET	/v3/users{?domain_id,name,enabled}	Lists users.
POST	/v3/users	Creates a user.
GET	/v3/users/{user_id}	Shows details for a user.
PATCH	/v3/users/{user_id}	Updates the password for or enables or disables a user.
DELETE	/v3/users/{user_id}	Deletes a user.
POST	/v3/users/{user_id}/password	Changes the password for a user.
GET	/v3/users/{user_id}/groups	Lists groups to which a user belongs.
GET	/v3/users/{user_id}/projects	List projects for a user.

7.11.1. List users

Method	URI	Description
GET	/v3/users{?domain_id,name,enabled}	Lists users.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.11.1.1. Request

This table shows the query parameters for the list users request:

Name	Type	Description
domain_id	UUID (Optional)	Filters the response by a domain ID.
name	String (Optional)	Filters the response by a user name.
enabled	String (Optional)	Filters the response by either enabled (true) or disabled (false) users.

This operation does not accept a request body.

7.11.1.2. Response

Example 7.88. List users: JSON response

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://localhost:5000/v3/users"
  },
  "users": [
    {
      "domain_id": "default",
      "email": null,
      "enabled": true,
      "id": "2844b2a08be147a08ef58317d6471f1f",
      "links": {
        "self": "http://localhost:5000/v3/users/2844b2a08be147a08ef58317d6471f1f"
      },
      "name": "glance"
    },
    {
      "domain_id": "default",
      "email": "test@example.com",
      "enabled": true,
      "id": "4ab84ab39de54f4d96eaff8f2145a7cd",
      "links": {
```



```

        "self": "http://localhost:5000/v3/users/
4ab84ab39de54f4d96eaff8f2145a7cd"
    },
    "name": "swiftusertest1"
},
{
    "domain_id": "default",
    "email": null,
    "enabled": true,
    "id": "56696a9a04864d63877a3d06a6f0b24b",
    "links": {
        "self": "http://localhost:5000/v3/users/
56696a9a04864d63877a3d06a6f0b24b"
    },
    "name": "swift"
},
{
    "domain_id": "default",
    "email": "alt_demo@example.com",
    "enabled": true,
    "id": "5acb638d15da44fc8de41b9a4bd41875",
    "links": {
        "self": "http://localhost:5000/v3/users/
5acb638d15da44fc8de41b9a4bd41875"
    },
    "name": "alt_demo"
},
{
    "domain_id": "default",
    "email": "demo@example.com",
    "enabled": true,
    "id": "7596e862b1af473c8ed6ae99d35b51e3",
    "links": {
        "self": "http://localhost:5000/v3/users/
7596e862b1af473c8ed6ae99d35b51e3"
    },
    "name": "demo"
},
{
    "domain_id": "default",
    "email": null,
    "enabled": true,
    "id": "802edb2141b44e77bbde241417450749",
    "links": {
        "self": "http://localhost:5000/v3/users/
802edb2141b44e77bbde241417450749"
    },
    "name": "nova"
},
{
    "domain_id": "592ab0800d3745baaf45c610fa41950a",
    "email": "test4@example.com",
    "enabled": true,
    "id": "9aca3883784647fe9aff3a50d922489a",
    "links": {
        "self": "http://localhost:5000/v3/users/
9aca3883784647fe9aff3a50d922489a"
    },
    "name": "swiftusertest4"
},

```

```

    {
      "domain_id": "default",
      "email": "test3@example.com",
      "enabled": true,
      "id": "a1251b011f9345e68c2458b841152034",
      "links": {
        "self": "http://localhost:5000/v3/users/a1251b011f9345e68c2458b841152034"
      },
      "name": "swiftusertest3"
    },
    {
      "domain_id": "default",
      "email": null,
      "enabled": true,
      "id": "a43f46eb318041f6b712143862e3ad70",
      "links": {
        "self": "http://localhost:5000/v3/users/a43f46eb318041f6b712143862e3ad70"
      },
      "name": "neutron"
    },
    {
      "domain_id": "default",
      "email": null,
      "enabled": true,
      "id": "b964a9e51c0046a4a84d3f83a135a97c",
      "links": {
        "self": "http://localhost:5000/v3/users/b964a9e51c0046a4a84d3f83a135a97c"
      },
      "name": "admin"
    },
    {
      "domain_id": "default",
      "email": null,
      "enabled": true,
      "id": "dc87e591c0d247d5ac04e873bd8a1646",
      "links": {
        "self": "http://localhost:5000/v3/users/dc87e591c0d247d5ac04e873bd8a1646"
      },
      "name": "cinder"
    },
    {
      "domain_id": "default",
      "email": "glance-swift@example.com",
      "enabled": true,
      "id": "ed214dc1c2c6468b926c96eca6c8aee9",
      "links": {
        "self": "http://localhost:5000/v3/users/ed214dc1c2c6468b926c96eca6c8aee9"
      },
      "name": "glance-swift"
    },
    {
      "domain_id": "default",
      "email": "test2@example.com",
      "enabled": true,
      "id": "f4f6587b058a4f46a00242549b430d37",

```

```
        "links": {
          "self": "http://localhost:5000/v3/users/
f4f6587b058a4f46a00242549b430d37"
        },
        "name": "swiftusertest2"
      }
    ]
  }
```

7.11.2. Create user

Method	URI	Description
POST	/v3/users	Creates a user.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.11.2.1. Request

Example 7.89. Create user: JSON request

```
{
  "user": {
    "default_project_id": "263fd9",
    "description": "James Doe user",
    "domain_id": "1789d1",
    "email": "jdoe@example.com",
    "enabled": true,
    "name": "James Doe",
    "password": "secretsecret"
  }
}
```

7.11.2.2. Response

Example 7.90. Create user: JSON response

```
{
  "user": {
    "default_project_id": "263fd9",
    "description": "James Doe user",
    "domain_id": "1789d1",
    "email": "jdoe@example.com",
    "enabled": true,
    "id": "ff4e51",
    "links": {
      "self": "https://identity:35357/v3/users/ff4e51"
    },
    "name": "James Doe"
  }
}
```

7.11.3. Show user details

Method	URI	Description
GET	/v3/users/{user_id}	Shows details for a user.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.11.3.1. Request

This table shows the URI parameters for the show user details request:

Name	Type	Description
{user_id}	UUID	The user ID.

This operation does not accept a request body.

7.11.3.2. Response

Example 7.91. Show user details: JSON response

```
{
  "user": {
    "default_project_id": "263fd9",
    "description": "John Smith's user",
    "domain_id": "1789d1",
    "email": "jsmith@example.com",
    "enabled": true,
    "id": "9fe1d3",
    "links": {
      "self": "https://identity:35357/v3/users/9fe1d3"
    },
    "name": "jsmith"
  }
}
```

7.11.4. Update user

Method	URI	Description
PATCH	/v3/users/{user_id}	Updates the password for or enables or disables a user.

If the back-end driver does not support this functionality, this call might return the HTTP Not Implemented (501) response code.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.11.4.1. Request

This table shows the URI parameters for the update user request:

Name	Type	Description
{user_id}	UUID	The user ID.

Example 7.92. Update user: JSON request

```
{
  "user": {
    "default_project_id": "263fd9",
    "description": "James Doe's user",
    "email": "jamesdoe@example.com",
    "enabled": true
  }
}
```

7.11.4.2. Response

Example 7.93. Update user: JSON response

```
{
  "user": {
    "default_project_id": "263fd9",
    "description": "James Doe's user",
    "domain_id": "1789d1",
    "email": "jamesdoe@example.com",
    "enabled": true,
    "id": "ff4e51",
    "links": {
      "self": "https://identity:35357/v3/users/ff4e51"
    },
    "name": "jamesdoe"
  }
}
```

7.11.5. Delete user

Method	URI	Description
DELETE	/v3/users/{user_id}	Deletes a user.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.11.5.1. Request

This table shows the URI parameters for the delete user request:

Name	Type	Description
{user_id}	UUID	The user ID.

This operation does not accept a request body.

7.11.6. Change password for user

Method	URI	Description
POST	/v3/users/{user_id}/password	Changes the password for a user.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

7.11.6.1. Request

This table shows the URI parameters for the change password for user request:

Name	Type	Description
{user_id}	UUID	The user ID.

Example 7.94. Change password for user: JSON request

```
{
  "user": {
    "password": "old_secretsecret",
    "original_password": "secretsecret"
  }
}
```


7.11.7. List groups to which a user belongs

Method	URI	Description
GET	/v3/users/{user_id}/groups	Lists groups to which a user belongs.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.11.7.1. Request

This table shows the URI parameters for the list groups to which a user belongs request:

Name	Type	Description
{user_id}	UUID	The user ID.

This operation does not accept a request body.

7.11.7.2. Response

Example 7.95. List groups to which a user belongs: JSON response

```
{
  "groups": [
    {
      "description": "Developers cleared for work on all general projects",
      "domain_id": "1789d1",
      "id": "ea167b",
      "links": {
        "self": "https://identity:35357/v3/groups/ea167b"
      },
      "name": "Developers"
    },
    {
      "description": "Developers cleared for work on secret projects",
      "domain_id": "1789d1",
      "id": "a62db1",
      "links": {
        "self": "https://identity:35357/v3/groups/a62db1"
      },
      "name": "Secure Developers"
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/users/9fe1d3/groups",
    "previous": null,
    "next": null
  }
}
```

7.11.8. List projects for user

Method	URI	Description
GET	/v3/users/{user_id}/projects	List projects for a user.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.11.8.1. Request

This table shows the URI parameters for the list projects for user request:

Name	Type	Description
{user_id}	UUID	The user ID.

This operation does not accept a request body.

7.11.8.2. Response

Example 7.96. List projects for user: JSON response

```
{
  "projects": [
    {
      "description": "description of this project",
      "domain_id": "161718",
      "enabled": true,
      "id": "456788",
      "parent_id": "212223",
      "links": {
        "self": "http://identity:35357/v3/projects/456788"
      },
      "name": "a project name"
    },
    {
      "description": "description of this project",
      "domain_id": "161718",
      "enabled": true,
      "id": "456789",
      "parent_id": "212223",
      "links": {
        "self": "http://identity:35357/v3/projects/456789"
      },
      "name": "another domain"
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/users/313233/projects",
    "previous": null,
    "next": null
  }
}
```

8. Identity API v3 extensions (CURRENT)

This page describes these Identity API v3 extensions:

- [OS-ENDPOINT-POLICY extension \(OS-ENDPOINT-POLICY\)](#)
- [OS-INHERIT extension \(OS-INHERIT\)](#)
- [Key Distribution Server \(KDS\) extension \(OS-KDS\)](#)
- [OAuth extension \(OS-OAUTH1\)](#)
- [Trust extension \(OS-TRUST\)](#)

Method	URI	Description
OS-ENDPOINT-POLICY extension (OS-ENDPOINT-POLICY)		
GET	/v3/policies/{policy_id}/OS-END-POINT-POLICY/endpoints	Lists all the endpoints that are currently associated with a policy through any of the association methods.
GET	/v3/policies/{policy_id}/OS-END-POINT-POLICY/policy	Shows a policy for an endpoint.
HEAD	/v3/policies/{policy_id}/OS-END-POINT-POLICY/policy	Checks whether a policy is associated with an endpoint.
PUT	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/end-points/{endpoint_id}	Associates a policy and an endpoint.
GET	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/end-points/{endpoint_id}	Verifies an association between a policy and an endpoint.
DELETE	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/end-points/{endpoint_id}	Deletes an association between a policy and an endpoint.
PUT	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/re-gions/{region_id}	Associates a policy and an endpoint of a service type in a region.
GET	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/re-gions/{region_id}	Verifies an association between a policy and service-type endpoint in a region.
DELETE	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/re-gions/{region_id}	Deletes an association between a policy and service-type endpoint in a region.
PUT	/v3/policies/{policy_id}/OS-END-POINT-POLICY/services/{service_id}	Associates a policy and any endpoint of a service type.
GET	/v3/policies/{policy_id}/OS-END-POINT-POLICY/services/{service_id}	Verifies an association between a policy and an endpoint of a service type.
DELETE	/v3/policies/{policy_id}/OS-END-POINT-POLICY/services/{service_id}	Deletes an association between a policy and an endpoint of a service type.
OS-INHERIT extension (OS-INHERIT)		
GET	/v3/role_assignments{?group.id, role.id, scope.domain.id, scope.project.id, user.id, effective}	Lists role assignments.
GET	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/inherited_to_projects	Lists the project roles that a user in a project inherits from a parent project.
PUT	/v3/OS-INHERIT/projects/{project_id}/users/	Assigns a role to a user in projects in a subtree.

Method	URI	Description
	{user_id}/roles/{role_id}/inherited_to_projects	
HEAD	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Checks whether a user has a role assignment with the inherited_to_projects flag in a project.
DELETE	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited role from a user in a project.
GET	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/inherited_to_projects	Lists the project roles that a group in a project inherits from a parent project.
PUT	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a group in projects in a subtree.
HEAD	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Checks whether a group has a role assignment with the inherited_to_projects flag in a project.
DELETE	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited role from a group in a project.
GET	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/inherited_to_projects	Lists the project roles that a user inherits from a parent project in a domain.
PUT	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a user in projects owned by a domain.
HEAD	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Checks whether a user has an inherited project role in a domain.
DELETE	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited project role from a user in a domain.
GET	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/inherited_to_projects	Lists the project roles that a group inherits from a parent project in a domain.
PUT	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a group in projects owned by a domain.
HEAD	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Checks whether a group has an inherited project role in a domain.
DELETE	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited project role from a group in a domain.
Key Distribution Server (KDS) extension (OS-KDS)		
POST	/v3/keys/{name}	Creates a long-term key in the KDS.

Method	URI	Description
DELETE	/v3/keys/{name}	Deletes a long-term key from the KDS.
POST	/v3/tickets	Generates a ticket to facilitate messaging between a source and destination.
GET	/v3/groups	Shows the key for a group in the KDS.
POST	/v3/groups/{name}	Creates a group in the KDS.
DELETE	/v3/groups/{name}	Deletes a group from the KDS.
OAuth extension (OS-OAUTH1)		
POST	/v3/OS-OAUTH1/consumers	Enables a user to create a consumer.
GET	/v3/OS-OAUTH1/consumers	Lists consumers.
GET	/v3/OS-OAUTH1/consumers/{consumer_id}	Shows details for a consumer.
PATCH	/v3/OS-OAUTH1/consumers/{consumer_id}	Updates the description for a consumer.
DELETE	/v3/OS-OAUTH1/consumers/{consumer_id}	Deletes a consumer.
POST	/v3/OS-OAUTH1/request_token	Enables a consumer to get an unauthorized request token.
POST	/v3/OS-OAUTH1/access_token	Enables a consumer to create an access token by exchanging a request token for an access token.
GET	/v3/OS-OAUTH1/users/{user_id}/access_tokens	Lists authorized access tokens.
GET	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}	Gets an authorized access token.
DELETE	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}	Enables a user to revoke an access token, which prevents the consumer from requesting new Identity Service API tokens. Also, revokes any Identity Service API tokens that were issued to the consumer through that access token.
GET	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}/roles	Lists associated roles for an access token.
GET	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}/roles/{role_id}	Shows details for a role for an access token.
POST	/v3/auth/tokens	Enables a consumer to get an Identity Service authentication token.
Trust extension (OS-TRUST)		
POST	/v3/OS-TRUST/trusts	Creates a trust.

8.1. OS-ENDPOINT-POLICY extension (OS-END-POINT-POLICY)

Creates, verifies, and deletes associations between service endpoints and policies. Such associations enable an endpoint to request its policy.

To create, check, or delete an association, you reference a policy by its ID in the Identity server.

The extension supports these associations:

1. A policy and endpoint association.
2. A policy and service-type endpoint in a region association.

3. A policy and service-type endpoint association.

This order reflects policies in their most to least-specific order.

When an endpoint requests the appropriate policy for itself, the extension finds the policy by traversing the ordered sequence of methods of association. The extension shows the policy for the first association that it finds.

If the region of the endpoint has a parent, the extension examines the region associations up the region tree in ascending order. For region associations, the extension examines any parent regions in ascending order. The extension does not combine policies.

Method	URI	Description
GET	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/endpoints	Lists all the endpoints that are currently associated with a policy through any of the association methods.
GET	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/policy	Shows a policy for an endpoint.
HEAD	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/policy	Checks whether a policy is associated with an endpoint.
PUT	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/endpoints/{endpoint_id}	Associates a policy and an endpoint.
GET	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/endpoints/{endpoint_id}	Verifies an association between a policy and an endpoint.
DELETE	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/endpoints/{endpoint_id}	Deletes an association between a policy and an endpoint.
PUT	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/regions/{region_id}	Associates a policy and an endpoint of a service type in a region.
GET	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/regions/{region_id}	Verifies an association between a policy and service-type endpoint in a region.
DELETE	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/regions/{region_id}	Deletes an association between a policy and service-type endpoint in a region.
PUT	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/{service_id}	Associates a policy and any endpoint of a service type.
GET	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/{service_id}	Verifies an association between a policy and an endpoint of a service type.
DELETE	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/{service_id}	Deletes an association between a policy and an endpoint of a service type.

8.1.1. List policy and service endpoint associations

Method	URI	Description
GET	/v3/policies/{policy_id}/OS-END-POINT-POLICY/endpoints	Lists all the endpoints that are currently associated with a policy through any of the association methods.

Normal response codes: 200

8.1.1.1. Request

This table shows the URI parameters for the list policy and service endpoint associations request:

Name	Type	Description
{policy_id}	UUID	The policy ID.

This operation does not accept a request body.

8.1.1.2. Response

Example 8.1. List policy and service endpoint associations: JSON response

```
{
  "endpoints": [
    {
      "id": "1",
      "interface": "public",
      "links": {
        "self": "http://identity:35357/v3/endpoints/1"
      },
      "region": "north",
      "service_id": "9242e05f0c23467bbd1cf1f7a6e5e596",
      "url": "http://identity:35357/"
    },
    {
      "id": "1",
      "interface": "internal",
      "links": {
        "self": "http://identity:35357/v3/endpoints/1"
      },
      "region": "south",
      "service_id": "9242e05f0c23467bbd1cf1f7a6e5e596",
      "url": "http://identity:35357/"
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "http://identity:35357/v3/OS-ENDPOINT-POLICY/policies/13c92821e4c4476a878d3aae7444f52f/endpoints"
  }
}
```

8.1.2. Show policy for endpoint

Method	URI	Description
GET	/v3/policies/{policy_id}/OS-END-POINT-POLICY/policy	Shows a policy for an endpoint.

The extension finds the policy by traversing the ordered sequence of methods of association. The extension shows the policy for the first association that it finds. If the region of the endpoint has a parent, the extension examines the region associations up the region tree in ascending order.

Normal response codes: 200

8.1.2.1. Request

This table shows the URI parameters for the show policy for endpoint request:

Name	Type	Description
{policy_id}	UUID	The policy ID.

This operation does not accept a request body.

8.1.2.2. Response

Example 8.2. Show policy for endpoint: JSON response

```
{
  "policy": {
    "blob": {
      "foobar_user": [
        "role:compute-user"
      ]
    },
    "id": "13c92821e4c4476a878d3aae7444f52f",
    "links": {
      "self": "http://identity:35357/v3/policies/13c92821e4c4476a878d3aae7444f52f"
    },
    "type": "application/json"
  }
}
```


8.1.3. Check policy and service endpoint association

Method	URI	Description
HEAD	/v3/policies/{policy_id}/OS-END-POINT-POLICY/policy	Checks whether a policy is associated with an endpoint.

Normal response codes: 200

8.1.3.1. Request

This table shows the URI parameters for the check policy and service endpoint association request:

Name	Type	Description
{policy_id}	UUID	The policy ID.

This operation does not accept a request body.

8.1.4. Associate policy and endpoint

Method	URI	Description
PUT	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/end-points/{endpoint_id}	Associates a policy and an endpoint.

If an association already exists between the endpoint and another policy, this call replaces that association.

Normal response codes: 204

8.1.4.1. Request

This table shows the URI parameters for the associate policy and endpoint request:

Name	Type	Description
{policy_id}	UUID	The policy ID.
{endpoint_id}	UUID	The endpoint ID.

This operation does not accept a request body.

8.1.5. Verify a policy and endpoint association

Method	URI	Description
GET	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/end-points/{endpoint_id}	Verifies an association between a policy and an endpoint.

A **HEAD** version of this API is also supported.

Normal response codes: 204

8.1.5.1. Request

This table shows the URI parameters for the verify a policy and endpoint association request:

Name	Type	Description
{policy_id}	UUID	The policy ID.
{endpoint_id}	UUID	The endpoint ID.

This operation does not accept a request body.

8.1.6. Delete a policy and endpoint association

Method	URI	Description
DELETE	/v3/policies/{policy_id}/ OS-ENDPOINT-POLICY/end- points/{endpoint_id}	Deletes an association between a policy and an endpoint.

Normal response codes: 204

8.1.6.1. Request

This table shows the URI parameters for the delete a policy and endpoint association request:

Name	Type	Description
{policy_id}	UUID	The policy ID.
{endpoint_id}	UUID	The endpoint ID.

This operation does not accept a request body.

8.1.7. Associate policy and service-type endpoint in a region

Method	URI	Description
PUT	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/regions/{region_id}	Associates a policy and an endpoint of a service type in a region.

If an association already exists between the service in a region and another policy, this call replaces that association.

Normal response codes: 204

8.1.7.1. Request

This table shows the URI parameters for the associate policy and service-type endpoint in a region request:

Name	Type	Description
{policy_id}	UUID	The policy ID.
{region_id}	UUID	The region ID.

This operation does not accept a request body.

8.1.8. Verify a policy and service-type endpoint in a region association

Method	URI	Description
GET	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/regions/{region_id}	Verifies an association between a policy and service-type endpoint in a region.

A HEAD version of this API is also supported.

Normal response codes: 204

8.1.8.1. Request

This table shows the URI parameters for the verify a policy and service-type endpoint in a region association request:

Name	Type	Description
{policy_id}	UUID	The policy ID.
{region_id}	UUID	The region ID.

This operation does not accept a request body.

8.1.9. Delete a policy and service-type endpoint in a region association

Method	URI	Description
DELETE	/v3/policies/{policy_id}/OS-ENDPOINT-POLICY/services/regions/{region_id}	Deletes an association between a policy and service-type endpoint in a region.

Normal response codes: 204

8.1.9.1. Request

This table shows the URI parameters for the delete a policy and service-type endpoint in a region association request:

Name	Type	Description
{policy_id}	UUID	The policy ID.
{region_id}	UUID	The region ID.

This operation does not accept a request body.

8.1.10. Associate policy and service-type endpoint

Method	URI	Description
PUT	/v3/policies/{policy_id}/OS-END-POINT-POLICY/services/{service_id}	Associates a policy and any endpoint of a service type.

If an association already exists between the endpoint of a service type and another policy, this call replaces that association.

Normal response codes: 204

8.1.10.1. Request

This table shows the URI parameters for the associate policy and service-type endpoint request:

Name	Type	Description
{policy_id}	UUID	The policy ID.
{service_id}	UUID	The service ID.

This operation does not accept a request body.

8.1.11. Verify a policy and service-type endpoint association

Method	URI	Description
GET	/v3/policies/{policy_id}/OS-END-POINT-POLICY/services/{service_id}	Verifies an association between a policy and an endpoint of a service type.

A HEAD version of this API is also supported.

Normal response codes: 204

8.1.11.1. Request

This table shows the URI parameters for the verify a policy and service-type endpoint association request:

Name	Type	Description
{policy_id}	UUID	The policy ID.
{service_id}	UUID	The service ID.

This operation does not accept a request body.

8.1.12. Delete a policy and service-type endpoint association

Method	URI	Description
DELETE	/v3/policies/{policy_id}/OS-END-POINT-POLICY/services/{service_id}	Deletes an association between a policy and an endpoint of a service type.

Normal response codes: 204

8.1.12.1. Request

This table shows the URI parameters for the delete a policy and service-type endpoint association request:

Name	Type	Description
{policy_id}	UUID	The policy ID.
{service_id}	UUID	The service ID.

This operation does not accept a request body.

8.2. OS-INHERIT extension (OS-INHERIT)

Enables projects to inherit role assignments from either their owning domain or projects that are higher in the hierarchy.

(Since API v3.4) The OS-INHERIT extension allows inheritance from both projects and domains. To access project inheritance, the Identity service server must run at least API v3.4.

Method	URI	Description
GET	/v3/role_assignments{?group.id, role.id, scope.domain.id, scope.project.id, user.id, effective}	Lists role assignments.
GET	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/inherited_to_projects	Lists the project roles that a user in a project inherits from a parent project.
PUT	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a user in projects in a subtree.
HEAD	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Checks whether a user has a role assignment with the inherited_to_projects flag in a project.
DELETE	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited role from a user in a project.
GET	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/inherited_to_projects	Lists the project roles that a group in a project inherits from a parent project.
PUT	/v3/OS-INHERIT/projects/{project_id}/groups/	Assigns a role to a group in projects in a subtree.

Method	URI	Description
	{group_id}/roles/{role_id}/inherited_to_projects	
HEAD	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Checks whether a group has a role assignment with the inherited_to_projects flag in a project.
DELETE	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited role from a group in a project.
GET	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/inherited_to_projects	Lists the project roles that a user inherits from a parent project in a domain.
PUT	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a user in projects owned by a domain.
HEAD	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Checks whether a user has an inherited project role in a domain.
DELETE	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited project role from a user in a domain.
GET	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/inherited_to_projects	Lists the project roles that a group inherits from a parent project in a domain.
PUT	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a group in projects owned by a domain.
HEAD	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Checks whether a group has an inherited project role in a domain.
DELETE	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited project role from a group in a domain.

8.2.1. List role assignments

Method	URI	Description
GET	/v3/role_assignments{?group.id,role.id,scope.domain.id,scope.project.id,user.id,effective}	Lists role assignments.

The scope section in the list response is extended to allow the representation of role assignments that are inherited to projects.

The list of all role assignments can be long. To filter the list, use the query parameters.

Some typical examples are:

GET /role_assignments?user.id={user_id} lists all role assignments for a user.

GET /role_assignments?scope.project.id={project_id} lists all role assignments for a project.

Each role assignment entity in the collection contains a link to the assignment that created this entity.

Use the `effective` query parameter to list effective assignments at the user, project, and domain level. This parameter allows for the effects of group membership as well as inheritance from the parent domain or project, for role assignments that were made using OS-INHERIT assignment APIs.

The group role assignment entities themselves are not returned in the collection. Because, like group membership, the effects of inheritance have already been allowed for, the role assignment entities themselves that specify the inheritance are not returned in the collection. This represents the effective role assignments that would be included in a scoped token. You can use the other query parameters with the `effective` parameter.

For example, to determine what a user can actually do, issue this request: GET /role_assignments?user.id={user_id}&effective

To get the equivalent set of role assignments that would be included in the token response of a project-scoped token, issue GET /role_assignments?user.id={user_id}&scope.project.id={project_id}&effective

In the response, the entity `links` section for entities that are included by virtue of group members also contains a url that you can use to access the membership of the group.

Use the `scope.OS-INHERIT:inherited_to` query parameter to filter the response by inherited role assignments. The `scope.OS-INHERIT:inherited_to` value of `projects` is currently supported. This value indicates that this role is inherited to all projects of the owning domain or parent project.

An example response for an API call with the `effective` query string:

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

8.2.1.1. Request

This operation does not accept a request body.

8.2.1.2. Response

Example 8.3. List role assignments: JSON response

```
{
  "role_assignments": [
    {
      "links": {
        "assignment": "http://identity:35357/v3/OS-INHERIT/domains/1234/users/5678/roles/91011/inherited_to_projects"
      },
      "role": {
        "id": "91011"
      },
      "scope": {
        "domain": {
          "id": "1234"
        },
        "OS-INHERIT:inherited_to": "projects"
      },
      "user": {
        "id": "5678"
      }
    },
    {
      "group": {
        "id": "5678"
      },
      "links": {
        "assignment": "http://identity:35357/v3/projects/1234/groups/5678/roles/91011"
      },
      "role": {
        "id": "91011"
      },
      "scope": {
        "project": {
          "id": "1234"
        }
      }
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/role_assignments",
    "previous": null,
    "next": null
  }
}
```

Example 8.4. List role assignments: JSON response

```
{
  "role_assignments": [
    {
      "links": {
        "assignment": "http://identity:35357/v3/OS-INHERIT/domains/12131/users/41516/roles/91011/inherited_to_projects"
      },
      "role": {
        "id": "91011"
      },
      "scope": {
        "project": {
          "id": "1234"
        }
      },
      "user": {
        "id": "41516"
      }
    },
    {
      "links": {
        "assignment": "http://identity:35357/v3/projects/1234/groups/5678/roles/91011",
        "membership": "http://identity:35357/v3/groups/5678/users/41516"
      },
      "role": {
        "id": "91011"
      },
      "scope": {
        "project": {
          "id": "1234"
        }
      },
      "user": {
        "id": "41516"
      }
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/role_assignments?effective",
    "previous": null,
    "next": null
  }
}
```

8.2.2. List roles for user

Method	URI	Description
GET	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/inherited_to_projects	Lists the project roles that a user in a project inherits from a parent project.

The list shows only roles that the user project inherits from the parent project.

Normal response codes: 200

8.2.2.1. Request

This table shows the URI parameters for the list roles for user request:

Name	Type	Description
{project_id}	UUID	The UUID of the project.
{user_id}	UUID	The UUID of the user.

This operation does not accept a request body.

8.2.2.2. Response

Example 8.5. List roles for user: JSON response

```
{
  "roles": [
    {
      "id": "91011",
      "links": {
        "self": "http://identity:35357/v3/roles/91011"
      },
      "name": "admin"
    },
    {
      "id": "91011",
      "links": {
        "self": "http://identity:35357/v3/roles/91011"
      },
      "name": "admin"
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/OS-INHERIT/projects/1234/users/5678/roles/inherited_to_projects",
    "previous": null,
    "next": null
  }
}
```

8.2.3. Assign role to user

Method	URI	Description
PUT	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a user in projects in a subtree.

The API anchors the inherited role assignment to a project and applies it to its subtree in the projects hierarchy to both existing and future projects.

A user can have both a regular, non-inherited role assignment and an inherited role assignment in the same project.

Normal response codes: 204

8.2.3.1. Request

This table shows the URI parameters for the assign role to user request:

Name	Type	Description
{project_id}	UUID	The UUID of the project.
{user_id}	UUID	The UUID of the user.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.2.4. Check role for user

Method	URI	Description
HEAD	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Checks whether a user has a role assignment with the inherited_to_projects flag in a project.

Normal response codes: 200

8.2.4.1. Request

This table shows the URI parameters for the check role for user request:

Name	Type	Description
{project_id}	UUID	The UUID of the project.
{user_id}	UUID	The UUID of the user.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.2.5. Revoke role from user

Method	URI	Description
DELETE	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited role from a user in a project.

Normal response codes: 204

8.2.5.1. Request

This table shows the URI parameters for the revoke role from user request:

Name	Type	Description
{project_id}	UUID	The UUID of the project.
{user_id}	UUID	The UUID of the user.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.2.6. List roles for group

Method	URI	Description
GET	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/inherited_to_projects	Lists the project roles that a group in a project inherits from a parent project.

The list shows only roles that the group project inherits from the parent project.

Normal response codes: 200

8.2.6.1. Request

This table shows the URI parameters for the list roles for group request:

Name	Type	Description
{project_id}	UUID	The UUID of the project.
{group_id}	UUID	The UUID of the group.

This operation does not accept a request body.

8.2.6.2. Response

Example 8.6. List roles for group: JSON response

```
{
  "roles": [
    {
      "id": "91011",
      "links": {
        "self": "http://identity:35357/v3/roles/91011"
      },
      "name": "admin"
    },
    {
      "id": "91011",
      "links": {
        "self": "http://identity:35357/v3/roles/91011"
      },
      "name": "admin"
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/OS-INHERIT/projects/1234/groups/5678/roles/inherited_to_projects",
    "previous": null,
    "next": null
  }
}
```

8.2.7. Assign role to group

Method	URI	Description
PUT	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a group in projects in a subtree.

The API anchors the inherited role assignment to a project and applies it to its subtree in the projects hierarchy to both existing and future projects.

A group can have both a regular, non-inherited role assignment and an inherited role assignment in the same project.

Normal response codes: 204

8.2.7.1. Request

This table shows the URI parameters for the assign role to group request:

Name	Type	Description
{project_id}	UUID	The UUID of the project.
{group_id}	UUID	The UUID of the group.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.2.8. Check role for group

Method	URI	Description
HEAD	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Checks whether a group has a role assignment with the inherited_to_projects flag in a project.

Normal response codes: 200

8.2.8.1. Request

This table shows the URI parameters for the check role for group request:

Name	Type	Description
{project_id}	UUID	The UUID of the project.
{group_id}	UUID	The UUID of the group.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.2.9. Revoke role from group

Method	URI	Description
DELETE	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited role from a group in a project.

Normal response codes: 204

8.2.9.1. Request

This table shows the URI parameters for the revoke role from group request:

Name	Type	Description
{project_id}	UUID	The UUID of the project.
{group_id}	UUID	The UUID of the group.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.2.10. List project roles for user in domain

Method	URI	Description
GET	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/inherited_to_projects	Lists the project roles that a user inherits from a parent project in a domain.

Normal response codes: 200

8.2.10.1. Request

This table shows the URI parameters for the list project roles for user in domain request:

Name	Type	Description
{domain_id}	UUID	The UUID of the domain.
{user_id}	UUID	The UUID of the user.

This operation does not accept a request body.

8.2.10.2. Response

Example 8.7. List project roles for user in domain: JSON response

```
{
  "roles": [
    {
      "id": "91011",
      "links": {
        "self": "http://identity:35357/v3/roles/91011"
      },
      "name": "admin"
    },
    {
      "id": "91011",
      "links": {
        "self": "http://identity:35357/v3/roles/91011"
      },
      "name": "admin"
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/OS-INHERIT/domains/1234/users/5678/roles/inherited_to_projects",
    "previous": null,
    "next": null
  }
}
```

8.2.11. Assign role to user owned by domain projects

Method	URI	Description
PUT	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a user in projects owned by a domain.

The API applies the inherited role to the existing and future owned projects. The inherited role does not appear as a role in a domain-scoped token.

Normal response codes: 204

8.2.11.1. Request

This table shows the URI parameters for the assign role to user owned by domain projects request:

Name	Type	Description
{domain_id}	UUID	The UUID of the domain.
{user_id}	UUID	The UUID of the user.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.2.12. Check project role for user in domain

Method	URI	Description
HEAD	/v3/OS-INHERIT/do-mains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Checks whether a user has an inherited project role in a domain.

Normal response codes: 200

8.2.12.1. Request

This table shows the URI parameters for the check project role for user in domain request:

Name	Type	Description
{domain_id}	UUID	The UUID of the domain.
{user_id}	UUID	The UUID of the user.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.2.13. Revoke project role from user in domain

Method	URI	Description
DELETE	/v3/OS-INHERIT/do-mains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited project role from a user in a domain.

Normal response codes: 204

8.2.13.1. Request

This table shows the URI parameters for the revoke project role from user in domain request:

Name	Type	Description
{domain_id}	UUID	The UUID of the domain.
{user_id}	UUID	The UUID of the user.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.2.14. List project roles for group in domain

Method	URI	Description
GET	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/inherited_to_projects	Lists the project roles that a group inherits from a parent project in a domain.

Normal response codes: 200

8.2.14.1. Request

This table shows the URI parameters for the list project roles for group in domain request:

Name	Type	Description
{domain_id}	UUID	The UUID of the domain.
{group_id}	UUID	The UUID of the group.

This operation does not accept a request body.

8.2.14.2. Response

Example 8.8. List project roles for group in domain: JSON response

```
{
  "roles": [
    {
      "id": "91011",
      "links": {
        "self": "http://identity:35357/v3/roles/91011"
      },
      "name": "admin"
    },
    {
      "id": "91011",
      "links": {
        "self": "http://identity:35357/v3/roles/91011"
      },
      "name": "admin"
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/OS-INHERIT/domains/1234/groups/5678/roles/inherited_to_projects",
    "previous": null,
    "next": null
  }
}
```

8.2.15. Assign role to group in domain projects

Method	URI	Description
PUT	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a group in projects owned by a domain.

The API applies the inherited role to owned projects, both existing and future. The inherited role does not appear as a role in a domain-scoped token.

Normal response codes: 204

8.2.15.1. Request

This table shows the URI parameters for the assign role to group in domain projects request:

Name	Type	Description
{domain_id}	UUID	The UUID of the domain.
{group_id}	UUID	The UUID of the group.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.2.16. Check project role for group in domain

Method	URI	Description
HEAD	/v3/OS-INHERIT/do-mains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Checks whether a group has an inherited project role in a domain.

Normal response codes: 200

8.2.16.1. Request

This table shows the URI parameters for the check project role for group in domain request:

Name	Type	Description
{domain_id}	UUID	The UUID of the domain.
{group_id}	UUID	The UUID of the group.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.2.17. Revoke project role from group in domain

Method	URI	Description
DELETE	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited project role from a group in a domain.

Normal response codes: 204

8.2.17.1. Request

This table shows the URI parameters for the revoke project role from group in domain request:

Name	Type	Description
{domain_id}	UUID	The UUID of the domain.
{group_id}	UUID	The UUID of the group.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.3. Key Distribution Server (KDS) extension (OS-KDS)

The Key Distribution Server (KDS) is a trusted third party that generates and securely distributes signing and encryption keys to communicating parties. These shared keys enable parties to exchange authenticated, confidential messages. KDS is an integral part of the RPC message security implementation. To establish a trusted relationship between the party and the KDS, a properly authorized user, such as a cloud administrator, must assign a long-term shared key to the communicating party. Assigning a key to a party requires assigning an identity to that party in the KDS. An identity includes a unique party name and the associated long-term shared key. This party name is used to identify a party when it communicates with KDS or another party.

The KDS enables two individual parties or one individual party and a group party to exchange secure messages. To get keys so that it can communicate with another party, a party makes an authenticated request to KDS for a ticket. The KDS returns an encrypted ticket to the requesting party. The KDS encrypts the ticket with the long-term shared key that is associated with that party. Only the associated party or the KDS itself can decrypt the ticket.

The KDS issues a ticket that contains a copy of the shared encryption and signing keys. These keys are for the source party, which is the party that requests the ticket. The ticket also contains a payload that is intended for the destination party, which is the party with whom the source party wants to communicate. This payload contains the information that the destination party can use to derive the shared encryption and signing keys. When the destination party is:

- An individual. The payload is encrypted with the long-term shared key that is associated with the destination party.
- A group. The payload is encrypted with a shared group key that the KDS makes available to all members of the group. This encryption enables the destination party to trust that the information in the payload was supplied by the KDS.

When the source party is ready to communicate with the destination party, it sends this encrypted payload to the destination party along with whatever data it has protected with the shared signing and encryption keys. The destination party can decrypt the payload and derive the shared encryption and signing keys by using the payload information. Both parties have a copy of the shared signing and encryption keys that are trusted as being issued by the KDS. The destination party uses these shared keys to authenticate and decrypt the data sent by the source party.

Other guidelines:

- When a source party must send secure messages to multiple recipients, an authorized user can define a KDS group for the recipients. Membership in a group is determined by comparing a party name with the group name. The party is considered a member if the party name matches `<group name> . *`. For example, a party named `scheduler . host . example . com` is considered a member of the `scheduler` group. This method is the same method that is used to name message queues in OpenStack.
- When a source party requests a ticket for a destination party that is a group, KDS generates a short-lived group key and assigns it to the group. This group key is used to encrypt the payload in the ticket, which contains the information that the destination party uses to derive the shared signing and encryption keys.
- When an individual destination party must decrypt the payload that it receives from the source party as a part of a group message, it makes an authenticated request to KDS to get the short-lived group key. If the requester is a member of the target group, KDS provides the short-lived group key encrypted with the long-term shared key associated with the individual destination party. The individual destination party can then decrypt the group key, which enables it to decrypt the payload and derive the shared signing and encryption keys to use to authenticate and decrypt the data sent by the source party.
- When a sender gets keys to send a message to a group, all group members and the sender share the signing and encryption keys, which makes it impossible for an individual destination party to determine whether a message was sent by the source party or by another destination party who is a group member. The destination party can confirm only that the message was sent by a party who has the shared signing and encryption keys. When a sender uses keys to send a message to a group, all group members must trust other members to operate in good faith.

The signing and encryption keys that communicating parties share are short-lived. The life span of these keys is defined by a validity period that is set by the KDS when it issues the ticket. A suggested reasonable default validity period is 15 minutes, though the implementation determines the appropriate validity period.

After the validity period for the keys expires, a party should refuse to use those keys to prevent the use of compromised keys. This requires the source party to request a new ticket from the KDS to get a new set of keys. An implementation can implement a grace period

to account for clock skew between parties. This grace period enables a destination party to accept messages that use recently expired keys. If you use a grace period, its duration should be five or fewer minutes.

A key server, unlike a pure public key-based system, can regulate the encryption and signing key exchange. When keys are requested, the key server actively distributes keys to communicating parties, applies access control, and denies communication between arbitrary peers in the system. This enables centralized access control, prevents unauthorized communication, and eliminates the need to perform post-authentication access control and policy lookups on the receiving side.

The KDS requires that all ticket requests are authenticated and, where appropriate, data is encrypted. You must pass any time stamp value to the API as a [UTC ISO 8601](#) date and time string that includes microseconds. For example, 2012-03-26T10:01:01.720000.

The default algorithms for message authentication and encryption are, respectively, HMAC-SHA-256 and AES-128-CBC. Therefore, the default block size is 128 bit.

The source party that requests a ticket must send the encrypted `esek` payload to the destination party. The source and destination strings used when requesting the ticket also must be sent to the destination party to enable it to derive the shared signing and encryption keys. The messaging implementation is responsible for transferring this data to the destination party.

The key derivation used to generate the shared signing and encryption keys uses the Hashed Message Authentication Code (HMAC)-based key derivation function (HKDF) standard, as described in RFC 5869. The destination party must use the HKDF expand function by using the information that it receives from the source party to complete derivation of the shared signing and encryption keys. The inputs to the HKDF expand function are:

```
HKDF-Expand(esek.key, info, 256)
```

The `info` input for the HKDF expand function is a string that concatenates the source, destination, and `esek.timestamp` strings by using a comma (,) separator between each element. The following example shows a valid `info` string where `scheduler.host.example.com` is the source, `compute.host.example.com` is the destination, and 2012-03-26T10:01:01.720000 is the `esek.timestamp`:

```
scheduler.host.example.com,compute.host.example.com,2012-03-26T10:01:01.720000
```

The output of the HKDF expand function is an array of bytes of 256 bit length. The first half is the signing key, and the second half is the encryption key.

Create and delete requests for long-term keys are restricted to authorized users, such as cloud administrators. The authentication and authorization for these requests is left up to the implementation, though the implementation should leverage the Identity API for these purposes.

Method	URI	Description
POST	/v3/keys/{name}	Creates a long-term key in the KDS.
DELETE	/v3/keys/{name}	Deletes a long-term key from the KDS.
POST	/v3/tickets	Generates a ticket to facilitate messaging between a source and destination.

Method	URI	Description
GET	/v3/groups	Shows the key for a group in the KDS.
POST	/v3/groups/{name}	Creates a group in the KDS.
DELETE	/v3/groups/{name}	Deletes a group from the KDS.

8.3.1. Create key

Method	URI	Description
POST	/v3/keys/{name}	Creates a long-term key in the KDS.

The request body contains the key.

The response shows the key name and generation value.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

8.3.1.1. Request

This table shows the URI parameters for the create key request:

Name	Type	Description
{name}	String	The name of the party that is associated with the key.

Example 8.9. Create key: JSON request

```
{
  "key": "TXkgcHJlY2lvdXNzcy4u..."
}
```

8.3.1.2. Response

Example 8.10. Create key: JSON response

```
{
  "name": "TXkgcHJlY2lvdXNzcy4u...",
  "generation": 2
}
```

8.3.2. Delete key

Method	URI	Description
DELETE	/v3/keys/{name}	Deletes a long-term key from the KDS.

Normal response codes: 204

8.3.2.1. Request

This table shows the URI parameters for the delete key request:

Name	Type	Description
{name}	String	The name of the party that is associated with the key.

This operation does not accept a request body.

8.3.3. Generate ticket

Method	URI	Description
POST	/v3/tickets	Generates a ticket to facilitate messaging between a source and destination.

A generate ticket request contains metadata that you specify as a Base64-encoded JSON object and a signature.

The response shows the metadata, encrypted ticket, and signature.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

8.3.3.1. Request

Example 8.11. Generate ticket: JSON request

```
{
  "metadata": "Zhn8yhasf8hikhf...",
  "signature": "c2lnbmF0dXJl..."
}
```

8.3.3.2. Response

Example 8.12. Generate ticket: JSON response

```
{
  "source": "scheduler.host.example.com",
  "destination": "compute.host.example.com",
  "timestamp": "2012-03-26T10:01:01.720000",
  "nonce": 1234567890
}
```

8.3.4. Show group key

Method	URI	Description
GET	/v3/groups	Shows the key for a group in the KDS.

When a ticket is requested where the destination is a group, a group key is generated that is valid for a predetermined amount of time. Any member of the group can get the key as long as it is still valid. Group keys are necessary to verify signatures and decrypt messages that have a group name as the target.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

8.3.4.1. Request

This operation does not accept a request body.

8.3.4.2. Response

Example 8.13. Show group key: JSON response

```
{
  "name": "--group-name--"
}
```

8.3.5. Create group

Method	URI	Description
POST	/v3/groups/{name}	Creates a group in the KDS.

Membership in groups is based on the party name. For example, a scheduler group implicitly includes any party name that starts with scheduler. For example, a member named scheduler.host.example.com.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

8.3.5.1. Request

This table shows the URI parameters for the create group request:

Name	Type	Description
{name}	String	The name of the group.

This operation does not accept a request body.

8.3.5.2. Response

Example 8.14. Create group: JSON response

```
{
  "name": "--group-name--"
}
```

8.3.6. Delete group

Method	URI	Description
DELETE	/v3/groups/{name}	Deletes a group from the KDS.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

8.3.6.1. Request

This table shows the URI parameters for the delete group request:

Name	Type	Description
{name}	String	The name of the group.

This operation does not accept a request body.

8.4. OAuth extension (OS-OAUTH1)

Enables users to delegate roles to third-party consumers through [The OAuth 1.0 Protocol](#).

A user is an Identity API user who delegates its roles and who authorizes request tokens.

A consumer is a third-party application that uses OAuth to access a protected resource.

Administrative users use an OAuth-derived token to act on behalf of the authorizing user.

The consumer uses a request token to get authorization from the user. The consumer provides an OAuth verifier string with the request token in exchange for an access token.

The consumer uses an access token to request Identity API tokens on behalf of the authorizing user instead of using the credentials for the user.

Request and access tokens use token keys to identify themselves. For OpenStack purposes, the token key is the token ID. The consumer uses a token secret to establish ownership of a token. Both request and access tokens have token secrets.

Delegated authentication through OAuth occurs as follows:

1. A user creates a consumer.
2. The consumer gets an unauthorized request token. Then, the consumer uses the request token to initiate user authorization.
3. The user authorizes the request token.
4. The consumer exchanges the authorized request token and the OAuth verifier for an access token.

The authorizing user receives the request token key from the consumer out-of-band.

5. The consumer uses the access token to request an Identity API token.

Method	URI	Description
POST	/v3/OS-0AUTH1/consumers	Enables a user to create a consumer.
GET	/v3/OS-0AUTH1/consumers	Lists consumers.
GET	/v3/OS-0AUTH1/consumers/{consumer_id}	Shows details for a consumer.
PATCH	/v3/OS-0AUTH1/consumers/{consumer_id}	Updates the description for a consumer.
DELETE	/v3/OS-0AUTH1/consumers/{consumer_id}	Deletes a consumer.
POST	/v3/OS-0AUTH1/request_token	Enables a consumer to get an unauthorized request token.
POST	/v3/OS-0AUTH1/access_token	Enables a consumer to create an access token by exchanging a request token for an access token.
GET	/v3/OS-0AUTH1/users/{user_id}/access_tokens	Lists authorized access tokens.
GET	/v3/OS-0AUTH1/users/{user_id}/access_tokens/{access_token_id}	Gets an authorized access token.
DELETE	/v3/OS-0AUTH1/users/{user_id}/access_tokens/{access_token_id}	Enables a user to revoke an access token, which prevents the consumer from requesting new Identity Service API tokens. Also, revokes any Identity Service API tokens that were issued to the consumer through that access token.
GET	/v3/OS-0AUTH1/users/{user_id}/access_tokens/{access_token_id}/roles	Lists associated roles for an access token.
GET	/v3/OS-0AUTH1/users/{user_id}/access_tokens/{access_token_id}/roles/{role_id}	Shows details for a role for an access token.
POST	/v3/auth/tokens	Enables a consumer to get an Identity Service authentication token.

8.4.1. Create consumer

Method	URI	Description
POST	/v3/OS-0AUTH1/consumers	Enables a user to create a consumer.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

8.4.1.1. Request

Example 8.15. Create consumer: JSON request

```
{
  "consumer": {
    "description": "My consumer"
  }
}
```

8.4.1.2. Response

Example 8.16. Create consumer: JSON response

```
{
  "consumer": {
    "secret": "secretsecret",
    "description": "My consumer",
    "id": "7fea2d",
    "links": {
      "self": "http://identity:35357/v3/OS-0AUTH1/consumers/7fea2d"
    }
  }
}
```

8.4.2. List consumers

Method	URI	Description
GET	/v3/OS-0AUTH1/consumers	Lists consumers.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

8.4.2.1. Request

This operation does not accept a request body.

8.4.2.2. Response

Example 8.17. List consumers: JSON response

```
{
  "consumers": [
    {
      "id": "0c2a74",
      "links": {
        "self": "http://identity:35357/v3/OS-0AUTH1/consumers/0c2a74"
      }
    },
    {
      "description": "My consumer",
      "id": "7fea2d",
      "links": {
        "self": "http://identity:35357/v3/OS-0AUTH1/consumers/7fea2d"
      }
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "http://identity:35357/v3/OS-0AUTH1/consumers"
  }
}
```

8.4.3. Show consumer details

Method	URI	Description
GET	/v3/OS-0AUTH1/consumers/{consumer_id}	Shows details for a consumer.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

8.4.3.1. Request

This table shows the URI parameters for the show consumer details request:

Name	Type	Description
{consumer_id}	UUID	The UUID of the consumer.

This operation does not accept a request body.

8.4.3.2. Response

Example 8.18. Show consumer details: JSON response

```
{
  "consumer": {
    "id": "7fea2d",
    "description": "My consumer",
    "links": {
      "self": "http://identity:35357/v3/OS-0AUTH1/consumers/7fea2d"
    }
  }
}
```

8.4.4. Update consumer

Method	URI	Description
PATCH	/v3/OS-0AUTH1/consumers/{consumer_id}	Updates the description for a consumer.

If you try to update any attribute other than description, the HTTP 400 Bad Request error is returned.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

8.4.4.1. Request

This table shows the URI parameters for the update consumer request:

Name	Type	Description
{consumer_id}	UUID	The UUID of the consumer.

Example 8.19. Update consumer: JSON request

```
{
  "consumer": {
    "description": "My new consumer"
  }
}
```

8.4.4.2. Response

Example 8.20. Update consumer: JSON response

```
{
  "consumer": {
    "description": "My new consumer",
    "id": "7fea2d",
    "links": {
      "self": "http://identity:35357/v3/OS-0AUTH1/consumers/7fea2d"
    }
  }
}
```

8.4.5. Delete consumer

Method	URI	Description
DELETE	/v3/OS-0AUTH1/consumers/{consumer_id}	Deletes a consumer.

When you delete a consumer, any associated request tokens, access tokens, and Identity API tokens are also deleted.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

8.4.5.1. Request

This table shows the URI parameters for the delete consumer request:

Name	Type	Description
{consumer_id}	UUID	The UUID of the consumer.

This operation does not accept a request body.

8.4.6. Show unauthorized request token

Method	URI	Description
POST	/v3/OS-OAUTH1/request_token	Enables a consumer to get an unauthorized request token.

Supported signature methods: HMAC-SHA1.

The consumer must provide all required OAuth parameters in the request. See [Consumer Obtains a Request Token](#).

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

8.4.6.1. Request

This operation does not accept a request body.

8.4.6.2. Response

Example 8.21. Show unauthorized request token: application/txtresponse

```
oauth_token=29971f&oauth_token_secret=238eb8&oauth_expires_at=
2013-09-11T06:07:51.501805Z
```

8.4.7. Create access token

Method	URI	Description
POST	/v3/OS-OAUTH1/access_token	Enables a consumer to create an access token by exchanging a request token for an access token.

After the user authorizes the request token, the consumer exchanges the authorized request token and OAuth verifier for an access token.

Supported signature methods: HMAC-SHA1.

The consumer must provide all required OAuth parameters in the request. See [Consumer Obtains a Request Token](#).

Supported signature methods: HMAC-SHA1.

You must provide all required OAuth parameters in the request. See [Consumer Obtains a Request Token](#).

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

8.4.7.1. Request

This operation does not accept a request body.

8.4.7.2. Response

Example 8.22. Create access token: application/txtresponse

```
oauth_token=accd36&oauth_token_secret=aa47da&oauth_expires_at=
2013-09-11T06:07:51.501805Z
```

8.4.8. List authorized access tokens

Method	URI	Description
GET	/v3/OS-0AUTH1/users/{user_id}/access_tokens	Lists authorized access tokens.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

8.4.8.1. Request

This table shows the URI parameters for the list authorized access tokens request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

This operation does not accept a request body.

8.4.8.2. Response

Example 8.23. List authorized access tokens: JSON response

```
{
  "access_tokens": [
    {
      "consumer_id": "7fea2d",
      "id": "6be26a",
      "expires_at": "2013-09-11T06:07:51.501805Z",
      "links": {
        "roles": "http://identity:35357/v3/users/ce9e07/OS-0AUTH1/
access_tokens/6be26a/roles",
        "self": "http://identity:35357/v3/users/ce9e07/OS-0AUTH1/
access_tokens/6be26a"
      },
      "project_id": "b9fca3",
      "authorizing_user_id": "ce9e07"
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "http://identity:35357/v3/users/ce9e07/OS-0AUTH1/
access_tokens"
  }
}
```


8.4.9. Get authorized access token

Method	URI	Description
GET	/v3/OS-0AUTH1/users/{user_id}/access_tokens/{access_token_id}	Gets an authorized access token.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

8.4.9.1. Request

This table shows the URI parameters for the get authorized access token request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.
{access_token_id}	UUID	The UUID of the access token.

This operation does not accept a request body.

8.4.9.2. Response

Example 8.24. Get authorized access token: JSON response

```
{
  "access_token": {
    "consumer_id": "7fea2d",
    "id": "6be26a",
    "expires_at": "2013-09-11T06:07:51.501805Z",
    "links": {
      "roles": "http://identity:35357/v3/users/ce9e07/OS-0AUTH1/
access_tokens/6be26a/roles",
      "self": "http://identity:35357/v3/users/ce9e07/OS-0AUTH1/
access_tokens/6be26a"
    },
    "project_id": "b9fca3",
    "authorizing_user_id": "ce9e07"
  }
}
```

8.4.10. Revoke access token

Method	URI	Description
DELETE	/v3/OS-0AUTH1/users/{user_id}/access_tokens/{access_token_id}	Enables a user to revoke an access token, which prevents the consumer from requesting new Identity Service API tokens. Also, revokes any Identity Service API tokens that were issued to the consumer through that access token.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

8.4.10.1. Request

This table shows the URI parameters for the revoke access token request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.
{access_token_id}	UUID	The UUID of the access token.

This operation does not accept a request body.

8.4.11. List roles for an access token

Method	URI	Description
GET	/v3/OS-0AUTH1/users/{user_id}/access_tokens/{access_token_id}/roles	Lists associated roles for an access token.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

8.4.11.1. Request

This table shows the URI parameters for the list roles for an access token request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.
{access_token_id}	UUID	The UUID of the access token.

This operation does not accept a request body.

8.4.12. Show role details for an access token

Method	URI	Description
GET	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}/roles/{role_id}	Shows details for a role for an access token.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

8.4.12.1. Request

This table shows the URI parameters for the show role details for an access token request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.
{access_token_id}	UUID	The UUID of the access token.
{role_id}	UUID	The UUID of the role.

This operation does not accept a request body.

8.4.12.2. Response

This operation does not return a response body.

8.4.13. Get an Identity Service token

Method	URI	Description
POST	/v3/auth/tokens	Enables a consumer to get an Identity Service authentication token.

The token represents the delegated authorization and identity (impersonation) of the authorizing user. The roles and scope of the generated token match those that the consumer initially requested.

Supported signature methods: HMAC-SHA1.

The consumer must provide required OAuth parameters in the request. See [Consumer Obtains a Request Token](#).

The returned token is scoped to the requested project and with the requested roles. In addition to the standard token response, the token has an OAuth-specific object.

Example OAuth-specific object in a token:

```
"OS-OAUTH1": {  
  "access_token_id": "cce0b8be7"  
}
```

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

8.4.13.1. Request

This operation does not accept a request body.

8.4.13.2. Response

This operation does not return a response body.

8.5. Trust extension (OS-TRUST)

Creates a trust.

A trust is an OpenStack Identity extension that enables delegation and, optionally, impersonation through keystone. A trust extension defines a relationship between a trustor and trustee. A trustor is the user who delegates a limited set of their own rights to another user, known as the trustee, for a limited time.

The trust can eventually enable the trustee to impersonate the trustor. For security reasons, some safety measures are added. For example, if a trustor loses a given role, the API automatically revokes any trusts and the related tokens that the user issued with that role.

For more information, see [Use trusts](#).

Method	URI	Description
POST	/v3/OS-TRUST/trusts	Creates a trust.

8.5.1. Create trust

Method	URI	Description
POST	/v3/OS-TRUST/trusts	Creates a trust.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), conflict (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

8.5.1.1. Request

Example 8.25. Create trust: JSON request

```
{
  "trust": {
    "expires_at": "2014-12-30T23:59:59.999999Z",
    "impersonation": false,
    "project_id": "'$PROJECT_ID'",
    "roles": [
      {
        "name": "admin"
      }
    ],
    "trustee_user_id": "'$DEMO_USER_ID'",
    "trustor_user_id": "'$ADMIN_USER_ID'"
  }
}
```

8.5.1.2. Response

Example 8.26. Create trust: JSON response

```
{
  "trust": {
    "expires_at": "2014-12-30T23:59:59.999999Z",
    "id": "394998fa61f14736b1f0c1f322882949",
    "impersonation": false,
    "links": {
      "self": "http://localhost:5000/v3/OS-TRUST/trusts/394998fa61f14736b1f0c1f322882949"
    },
    "project_id": "3d4c2c82bd5948f0bcab0cf3a7c9b48c",
    "remaining_uses": null,
    "roles": [
      {
        "id": "c703057be878458588961ce9a0ce686b",
        "links": {
          "self": "http://localhost:5000/v3/roles/c703057be878458588961ce9a0ce686b"
        },
        "name": "admin"
      }
    ]
  }
}
```

```
    "roles_links": {
      "next": null,
      "previous": null,
      "self": "http://localhost:5000/v3/OS-TRUST/trusts/
394998fa61f14736b1f0c1f322882949/roles"
    },
    "trustee_user_id": "269348fdd9374b8885da1418e0730af1",
    "trustor_user_id": "3ec3164f750146be97f21559ee4d9c51"
  }
}
```


9. Identity API v2.0 (SUPPORTED)

Gets an authentication token that permits access to the OpenStack services REST API.

Method	URI	Description
API versions		
GET	/	Lists information about all Identity API versions.
GET	/v2.0	Shows details for the Identity API v2.0.
Extensions		
GET	/v2.0/extensions	Lists available extensions.
GET	/v2.0/extensions/{alias}	Shows details for an extension, by alias.
Tokens and tenants		
POST	/v2.0/tokens	Authenticates and generates a token.
GET	/v2.0/tenants{?limit,marker}	Lists tenants to which the token has access.

9.1. API versions

Method	URI	Description
GET	/	Lists information about all Identity API versions.
GET	/v2.0	Shows details for the Identity API v2.0.

9.1.1. List versions

Method	URI	Description
GET	/	Lists information about all Identity API versions.

Normal response codes: 200300

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

9.1.1.1. Request

This operation does not accept a request body.

9.1.1.2. Response

Example 9.1. List versions: JSON response

```
{
  "versions": {
    "values": [
      {
        "id": "v3.4",
        "links": [
          {
            "href": "http://localhost:35357/v3/",
            "rel": "self"
          }
        ],
        "media-types": [
          {
            "base": "application/json",
            "type": "application/vnd.openstack.identity-v3+json"
          }
        ],
        "status": "stable",
        "updated": "2015-03-30T00:00:00Z"
      },
      {
        "id": "v2.0",
        "links": [
          {
            "href": "http://localhost:35357/v2.0/",
            "rel": "self"
          },
          {
            "href": "http://docs.openstack.org/",
            "rel": "describedby",
            "type": "text/html"
          }
        ],
        "media-types": [
          {
            "base": "application/json",
```

```
        "type": "application/vnd.openstack.identity-v2.0+json"
      },
    ],
    "status": "stable",
    "updated": "2014-04-17T00:00:00Z"
  }
]
}
```

9.1.2. Show version details

Method	URI	Description
GET	/v2.0	Shows details for the Identity API v2.0.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

9.1.2.1. Request

This operation does not accept a request body.

9.1.2.2. Response

Example 9.2. Show version information: JSON response

```
{
  "version": {
    "status": "stable",
    "updated": "2014-04-17T00:00:00Z",
    "media-types": [
      {
        "base": "application/json",
        "type": "application/vnd.openstack.identity-v2.0+json"
      }
    ],
    "id": "v2.0",
    "links": [
      {
        "href": "http://localhost:5000/v2.0/",
        "rel": "self"
      },
      {
        "href": "http://docs.openstack.org/",
        "rel": "describedby",
        "type": "text/html"
      }
    ]
  }
}
```

9.2. Extensions

Method	URI	Description
GET	/v2.0/extensions	Lists available extensions.
GET	/v2.0/extensions/{alias}	Shows details for an extension, by alias.

9.2.1. List extensions

Method	URI	Description
GET	/v2.0/extensions	Lists available extensions.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

9.2.1.1. Request

This operation does not accept a request body.

9.2.1.2. Response

Example 9.3. List extensions: JSON response

```
{
  "extensions": {
    "values": [
      {
        "updated": "2013-07-07T12:00:00-00:00",
        "name": "OpenStack S3 API",
        "links": [
          {
            "href": "https://github.com/openstack/identity-api",
            "type": "text/html",
            "rel": "describedby"
          }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/s3tokens/v1.0",
        "alias": "s3tokens",
        "description": "OpenStack S3 API."
      },
      {
        "updated": "2013-07-23T12:00:00-00:00",
        "name": "OpenStack Keystone Endpoint Filter API",
        "links": [
          {
            "href": "https://github.com/openstack/identity-api/blob/master/openstack-identity-api/v3/src/markdown/identity-api-v3-os-ep-filter-ext.md",
            "type": "text/html",
            "rel": "describedby"
          }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-EP-FILTER/v1.0",
        "alias": "OS-EP-FILTER",
        "description": "OpenStack Keystone Endpoint Filter API."
      },
      {
        "updated": "2014-02-24T20:51:00-00:00",
        "name": "OpenStack Revoke API",
```

```

        "links": [
            {
                "href": "https://github.com/openstack/identity-api/
blob/master/openstack-identity-api/v3/src/markdown/identity-api-v3-os-revoke-
ext.md",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-
REVOKE/v1.0",
        "alias": "OS-REVOKE",
        "description": "OpenStack revoked token reporting mechanism."
    },
    {
        "updated": "2013-12-17T12:00:0-00:00",
        "name": "OpenStack Federation APIs",
        "links": [
            {
                "href": "https://github.com/openstack/identity-api",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-
FEDERATION/v1.0",
        "alias": "OS-FEDERATION",
        "description": "OpenStack Identity Providers Mechanism."
    },
    {
        "updated": "2013-07-11T17:14:00-00:00",
        "name": "OpenStack Keystone Admin",
        "links": [
            {
                "href": "https://github.com/openstack/identity-api",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-
KSADM/v1.0",
        "alias": "OS-KSADM",
        "description": "OpenStack extensions to Keystone v2.0 API
enabling Administrative Operations."
    },
    {
        "updated": "2014-01-20T12:00:0-00:00",
        "name": "OpenStack Simple Certificate API",
        "links": [
            {
                "href": "https://github.com/openstack/identity-api",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-
SIMPLE-CERT/v1.0",
        "alias": "OS-SIMPLE-CERT",
        "description": "OpenStack simple certificate retrieval
extension"
    }

```

```

    },
    {
      "updated": "2013-07-07T12:00:0-00:00",
      "name": "OpenStack OAUTH1 API",
      "links": [
        {
          "href": "https://github.com/openstack/identity-api",
          "type": "text/html",
          "rel": "describedby"
        }
      ],
      "namespace": "http://docs.openstack.org/identity/api/ext/OS-
OAUTH1/v1.0",
      "alias": "OS-OAUTH1",
      "description": "OpenStack OAuth 1.0a Delegated Auth Mechanism.
"
    },
    {
      "updated": "2013-07-07T12:00:0-00:00",
      "name": "OpenStack EC2 API",
      "links": [
        {
          "href": "https://github.com/openstack/identity-api",
          "type": "text/html",
          "rel": "describedby"
        }
      ],
      "namespace": "http://docs.openstack.org/identity/api/ext/OS-
EC2/v1.0",
      "alias": "OS-EC2",
      "description": "OpenStack EC2 Credentials backend."
    }
  ]
}

```

9.2.2. Show extension details

Method	URI	Description
GET	/v2.0/extensions/{alias}	Shows details for an extension, by alias.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

9.2.2.1. Request

This table shows the URI parameters for the show extension details request:

Name	Type	Description
{alias}	String	An alias for the extension name. For example, os-server-external-events.

This operation does not accept a request body.

9.2.2.2. Response

Example 9.4. Show extension details: JSON response

```
{
  "extension": {
    "updated": "2013-07-07T12:00:0-00:00",
    "name": "OpenStack OAuth1 API",
    "links": [
      {
        "href": "https://github.com/openstack/identity-api",
        "type": "text/html",
        "rel": "describedby"
      }
    ],
    "namespace": "http://docs.openstack.org/identity/api/ext/OS-OAuth1/v1.0",
    "alias": "OS-OAuth1",
    "description": "OpenStack OAuth 1.0a Delegated Auth Mechanism."
  }
}
```

9.3. Tokens and tenants

Method	URI	Description
POST	/v2.0/tokens	Authenticates and generates a token.
GET	/v2.0/tenants{?limit,marker}	Lists tenants to which the token has access.

9.3.1. Authenticate

Method	URI	Description
POST	/v2.0/tokens	Authenticates and generates a token.

The Identity API is a RESTful web service. It is the entry point to all service APIs. To access the Identity API, you must know its URL.

Each REST request against Identity requires the X-Auth-Token header. Clients obtain this token, along with the URL to other service APIs, by first authenticating against Identity with valid credentials.

To authenticate, you must provide either a user ID and password or a token.

If the authentication token has expired, this call returns the HTTP 401 status code.

If the token has expired, this call returns the HTTP 404 status code.

The Identity API treats expired tokens as no longer valid tokens.

The deployment determines how long expired tokens are stored.

To view the trust object, you need to set trust enable on the keystone configuration.

Normal response codes: 200203

Error response codes: userDisabled (403), Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

9.3.1.1. Request

Example 9.5. Authenticate with user name and password credentials: JSON request

```
{
  "auth": {
    "tenantName": "demo",
    "passwordCredentials": {
      "username": "demo",
      "password": "secretsecret"
    }
  }
}
```

Example 9.6. Authenticate with token: JSON request

```
{
  "auth": {
    "tenantName": "demo",
    "token": {
      "id": "cbc36478b0bd8e67e89469c7749d4127"
    }
  }
}
```

9.3.1.2. Response

Example 9.7. Authenticate with user name and password credentials: JSON response

```
{
  "access": {
    "token": {
      "issued_at": "2014-01-30T15:30:58.819584",
      "expires": "2014-01-31T15:30:58Z",
      "id": "aaaaa-bbbbbb-cccc-dddd",
      "tenant": {
        "description": null,
        "enabled": true,
        "id": "fc394f2ab2df4114bde39905f800dc57",
        "name": "demo"
      }
    },
    "serviceCatalog": [
      {
        "endpoints": [
          {
            "adminURL": "http://23.253.72.207:8774/v2/
fc394f2ab2df4114bde39905f800dc57",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8774/v2/
fc394f2ab2df4114bde39905f800dc57",
            "id": "2dad48f09e2a447a9bf852bcd93548ef",
            "publicURL": "http://23.253.72.207:8774/v2/
fc394f2ab2df4114bde39905f800dc57"
          }
        ],
        "endpoints_links": [],
        "type": "compute",
        "name": "nova"
      },
      {
        "endpoints": [
          {
            "adminURL": "http://23.253.72.207:9696/",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:9696/",
            "id": "97c526db8d7a4c88bbb8d68db1bdcdb8",
            "publicURL": "http://23.253.72.207:9696/"
          }
        ],
        "endpoints_links": [],
        "type": "network",
        "name": "neutron"
      },
      {
        "endpoints": [
          {
            "adminURL": "http://23.253.72.207:8776/v2/
fc394f2ab2df4114bde39905f800dc57",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8776/v2/
fc394f2ab2df4114bde39905f800dc57",
            "id": "93f86dfcbba143a39a33d0c2cd424870",
```

```

        "publicURL": "http://23.253.72.207:8776/v2/
fc394f2ab2df4114bde39905f800dc57"
    },
    "endpoints_links": [],
    "type": "volumev2",
    "name": "cinder"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8774/v3",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8774/v3",
            "id": "3eb274b12b1d47b2abc536038d87339e",
            "publicURL": "http://23.253.72.207:8774/v3"
        }
    ],
    "endpoints_links": [],
    "type": "compute_v3",
    "name": "nova"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:3333",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:3333",
            "id": "957f1e54afc64d33a62099faa5e980a2",
            "publicURL": "http://23.253.72.207:3333"
        }
    ],
    "endpoints_links": [],
    "type": "s3",
    "name": "s3"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:9292",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:9292",
            "id": "27d5749f36864c7d96bebf84a5ec9767",
            "publicURL": "http://23.253.72.207:9292"
        }
    ],
    "endpoints_links": [],
    "type": "image",
    "name": "glance"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8776/v1/
fc394f2ab2df4114bde39905f800dc57",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8776/v1/
fc394f2ab2df4114bde39905f800dc57",
            "id": "37c83a2157f944f1972e74658aa0b139",

```

```

        "publicURL": "http://23.253.72.207:8776/v1/
fc394f2ab2df4114bde39905f800dc57"
    },
    ],
    "endpoints_links": [],
    "type": "volume",
    "name": "cinder"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8773/services/
Admin",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8773/services/
Cloud",
            "id": "289b59289d6048e2912b327e5d3240ca",
            "publicURL": "http://23.253.72.207:8773/services/
Cloud"
        }
    ],
    "endpoints_links": [],
    "type": "ec2",
    "name": "ec2"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8080",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8080/v1/
AUTH_fc394f2ab2df4114bde39905f800dc57",
            "id": "16b76b5e5b7d48039a6e4cc3129545f3",
            "publicURL": "http://23.253.72.207:8080/v1/
AUTH_fc394f2ab2df4114bde39905f800dc57"
        }
    ],
    "endpoints_links": [],
    "type": "object-store",
    "name": "swift"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:35357/v2.0",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:5000/v2.0",
            "id": "26af053673df4ef3a2340c4239e21ea2",
            "publicURL": "http://23.253.72.207:5000/v2.0"
        }
    ],
    "endpoints_links": [],
    "type": "identity",
    "name": "keystone"
}
],
"user": {
    "username": "demo",
    "roles_links": [],
    "id": "9a6590b2ab024747bc2167c4e064d00d",

```

```
    "roles": [
      {
        "name": "Member"
      },
      {
        "name": "anotherrole"
      }
    ],
    "name": "demo"
  },
  "metadata": {
    "is_admin": 0,
    "roles": [
      "7598ac3c634d4c3da4b9126a5f67ca2b",
      "f95c0ab82d6045d9805033ee1fbc80d4"
    ]
  },
  "trust": {
    "id": "394998fa61f14736b1f0c1f322882949",
    "trustee_user_id": "269348fdd9374b8885da1418e0730af1",
    "trustor_user_id": "3ec3164f750146be97f21559ee4d9c51",
    "impersonation": false
  }
}
```

9.3.2. List tenants

Method	URI	Description
GET	/v2.0/tenants{?limit,marker}	Lists tenants to which the token has access.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

9.3.2.1. Request

This table shows the header parameters for the list tenants request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token.

Example 9.8. List tenants: HTTP/JSON request

```
GET /v2.0/tenants HTTP/1.1
Host: identity.api.openstack.org
Content-Type: application/json
X-Auth-Token: fa8426a0-8eaf-4d22-8e13-7c1b16a9370c
Accept: application/json
```

This operation does not accept a request body.

9.3.2.2. Response

Example 9.9. List tenants: JSON response

```
{
  "tenants": [
    {
      "id": "1234",
      "name": "ACME Corp",
      "description": "A description ...",
      "enabled": true
    },
    {
      "id": "3456",
      "name": "Iron Works",
      "description": "A description ...",
      "enabled": true
    }
  ],
  "tenants_links": []
}
```

10. Identity Admin API v2.0 (SUPPORTED)

Gets an authentication token that permits access to the Compute API.

Method	URI	Description
Versions		
GET	/v2.0	Gets detailed information about a version of the Identity API.
Extensions		
Tokens		
POST	/v2.0/tokens	Authenticates and generates a token.
GET	/v2.0/tokens/{tokenId}{?belongsTo}	Validates a token and confirms that it belongs to a tenant.
HEAD	/v2.0/tokens/{tokenId}{?belongsTo}	Validates a token and confirms that it belongs to a tenant, for performance.
DELETE	/v2.0/tokens/{tokenId}{?belongsTo}	Deletes a token.
GET	/v2.0/tokens/{tokenId}/endpoints{?limit,marker}	Lists the endpoints associated with a token.
Users		
GET	/v2.0/users{?name}	Lists all users.
POST	/v2.0/users	Creates a user.
PUT	/v2.0/users/{userId}	Updates a user.
DELETE	/v2.0/users/{userId}	Deletes a user.
GET	/v2.0/users/{userId}	Shows details for a user, by ID.
GET	/v2.0/users/{userId}/roles{?limit,marker}	Lists global roles for a user. Excludes tenant roles.
Tenants		
GET	/v2.0/tenants{?limit,marker}	Lists all tenants.
GET	/v2.0/tenants{?name}	Shows details for a tenant, by name.
GET	/v2.0/tenants/{tenantId}	Shows details for a tenant, by ID.
GET	/v2.0/tenants/{tenantId}/users/{userId}/roles{?limit,marker}	Lists roles for a user on a tenant. Excludes global roles.

10.1. Versions

Method	URI	Description
GET	/v2.0	Gets detailed information about a version of the Identity API.

10.1.1. Get version details

Method	URI	Description
GET	/v2.0	Gets detailed information about a version of the Identity API.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.1.1.1. Request

This operation does not accept a request body.

10.1.1.2. Response

Example 10.1. Get version information: JSON response

```
{
  "version": {
    "status": "stable",
    "updated": "2014-04-17T00:00:00Z",
    "media-types": [
      {
        "base": "application/json",
        "type": "application/vnd.openstack.identity-v2.0+json"
      }
    ],
    "id": "v2.0",
    "links": [
      {
        "href": "http://localhost:5000/v2.0/",
        "rel": "self"
      },
      {
        "href": "http://docs.openstack.org/",
        "rel": "describedby",
        "type": "text/html"
      }
    ]
  }
}
```

10.2. Extensions

For information about Identity API v2.0 extensions, see [Extensions](#).

10.3. Tokens

Method	URI	Description
POST	/v2.0/tokens	Authenticates and generates a token.

Method	URI	Description
GET	/v2.0/tokens/{tokenId}{?belongsTo}	Validates a token and confirms that it belongs to a tenant.
HEAD	/v2.0/tokens/{tokenId}{?belongsTo}	Validates a token and confirms that it belongs to a tenant, for performance.
DELETE	/v2.0/tokens/{tokenId}{?belongsTo}	Deletes a token.
GET	/v2.0/tokens/{tokenId}/endpoints{?limit,marker}	Lists the endpoints associated with a token.

10.3.1. Authenticate for admin API

Method	URI	Description
POST	/v2.0/tokens	Authenticates and generates a token.

A REST interface provides client authentication by using the **POST** method with `v2.0/tokens` as the path. Include a payload of credentials in the body.

The Identity API is a RESTful web service. It is the entry point to all service APIs. To access the Identity API, you must know its URL.

Each REST request against the Identity Service requires the `X-Auth-Token` header. Clients obtain this token and the URL endpoints for other service APIs by supplying their valid credentials to the authentication service.

If the authentication token has expired, this call returns the HTTP unauthorized (401) response code.

If the token has expired, this call returns the `itemNotFound` (404) response code.

The Identity API treats expired tokens as no longer valid tokens.

The deployment determines how long expired tokens are stored.

To view the trust object, you need to set `trust enable` on the keystone configuration.

Normal response codes: 200203

Error response codes: `userDisabled` (403), `Bad Request` (400), `Unauthorized` (401), `Forbidden` (403), `Not Found` (404), `Method Not Allowed` (405), `Request Entity Too Large` (413), `Service Unavailable` (503)

10.3.1.1. Request

This table shows the header parameters for the authenticate for admin api request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

Example 10.2. Authenticate with credentials: JSON request

```
{
  "auth": {
    "tenantName": "admin",
    "passwordCredentials": {
      "username": "admin",
      "password": "secretsecret"
    }
  }
}
```

Example 10.3. Authenticate with token: JSON request

```
{
  "auth": {
```

```

        "tenantName": "demo",
        "token": {
            "id": "cbc36478b0bd8e67e89469c7749d4127"
        }
    }
}

```

10.3.1.2. Response

Example 10.4. Authenticate with credentials: JSON response

```

{
  "access": {
    "token": {
      "issued_at": "2014-01-30T17:09:57.647795",
      "expires": "2014-01-31T17:09:57Z",
      "id": "admin_id",
      "tenant": {
        "description": null,
        "enabled": true,
        "id": "73f0aa26640f4971864919d0eb0f0880",
        "name": "admin"
      }
    },
    "serviceCatalog": [
      {
        "endpoints": [
          {
            "adminURL": "http://23.253.72.207:8774/v2/73f0aa26640f4971864919d0eb0f0880",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8774/v2/73f0aa26640f4971864919d0eb0f0880",
            "id": "2dad48f09e2a447a9bf852bcd93548ef",
            "publicURL": "http://23.253.72.207:8774/v2/73f0aa26640f4971864919d0eb0f0880"
          }
        ],
        "endpoints_links": [],
        "type": "compute",
        "name": "nova"
      },
      {
        "endpoints": [
          {
            "adminURL": "http://23.253.72.207:9696/",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:9696/",
            "id": "97c526db8d7a4c88bbb8d68db1bdcdb8",
            "publicURL": "http://23.253.72.207:9696/"
          }
        ],
        "endpoints_links": [],
        "type": "network",
        "name": "neutron"
      },
      {
        "endpoints": [

```

```

        "adminURL": "http://23.253.72.207:8776/v2/
73f0aa26640f4971864919d0eb0f0880",
        "region": "RegionOne",
        "internalURL": "http://23.253.72.207:8776/v2/
73f0aa26640f4971864919d0eb0f0880",
        "id": "93f86dfcbba143a39a33d0c2cd424870",
        "publicURL": "http://23.253.72.207:8776/v2/
73f0aa26640f4971864919d0eb0f0880"
    }
},
{
    "endpoints_links": [],
    "type": "volumev2",
    "name": "cinder"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8774/v3",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8774/v3",
            "id": "3eb274b12b1d47b2abc536038d87339e",
            "publicURL": "http://23.253.72.207:8774/v3"
        }
    ],
    "endpoints_links": [],
    "type": "computev3",
    "name": "nova"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:3333",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:3333",
            "id": "957f1e54afc64d33a62099faa5e980a2",
            "publicURL": "http://23.253.72.207:3333"
        }
    ],
    "endpoints_links": [],
    "type": "s3",
    "name": "s3"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:9292",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:9292",
            "id": "27d5749f36864c7d96bebf84a5ec9767",
            "publicURL": "http://23.253.72.207:9292"
        }
    ],
    "endpoints_links": [],
    "type": "image",
    "name": "glance"
},
{
    "endpoints": [
        {

```

```

        "adminURL": "http://23.253.72.207:8776/v1/
73f0aa26640f4971864919d0eb0f0880",
        "region": "RegionOne",
        "internalURL": "http://23.253.72.207:8776/v1/
73f0aa26640f4971864919d0eb0f0880",
        "id": "37c83a2157f944f1972e74658aa0b139",
        "publicURL": "http://23.253.72.207:8776/v1/
73f0aa26640f4971864919d0eb0f0880"
    }
},
"endpoints_links": [],
"type": "volume",
"name": "cinder"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8773/services/
Admin",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8773/services/
Cloud",
            "id": "289b59289d6048e2912b327e5d3240ca",
            "publicURL": "http://23.253.72.207:8773/services/
cloud"
        }
    ],
    "endpoints_links": [],
    "type": "ec2",
    "name": "ec2"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8080",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8080/v1/
AUTH_73f0aa26640f4971864919d0eb0f0880",
            "id": "16b76b5e5b7d48039a6e4cc3129545f3",
            "publicURL": "http://23.253.72.207:8080/v1/
AUTH_73f0aa26640f4971864919d0eb0f0880"
        }
    ],
    "endpoints_links": [],
    "type": "object-store",
    "name": "swift"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:35357/v2.0",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:5000/v2.0",
            "id": "26af053673df4ef3a2340c4239e21ea2",
            "publicURL": "http://23.253.72.207:5000/v2.0"
        }
    ],
    "endpoints_links": [],
    "type": "identity",
    "name": "keystone"
}

```

```
    },
    ],
    "user": {
      "username": "admin",
      "roles_links": [],
      "id": "1f568815cb8148688e6ee9b2f7527dcc",
      "roles": [
        {
          "name": "service"
        },
        {
          "name": "admin"
        }
      ],
      "name": "admin"
    },
    "metadata": {
      "is_admin": 0,
      "roles": [
        "8341d3603a1d4d5985bfff09f10704d4d",
        "2e66d57df76946fdb034bc4da6fdec0"
      ]
    },
    "trust": {
      "id": "394998fa61f14736b1f0c1f322882949",
      "trustee_user_id": "269348fdd9374b8885da1418e0730af1",
      "trustor_user_id": "3ec3164f750146be97f21559ee4d9c51",
      "impersonation": false
    }
  }
}
```

10.3.2. Validate token

Method	URI	Description
GET	/v2.0/tokens/{tokenId}?belongsTo	Validates a token and confirms that it belongs to a tenant.

Returns the permissions relevant to a particular client. Valid tokens are in the /tokens/{tokenId} path. If the token is not valid, this call returns the `itemNotFound` (404) response code.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.3.2.1. Request

This table shows the header parameters for the validate token request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the validate token request:

Name	Type	Description
{tokenId}	UUID	The authentication token for which to perform the operation.

This operation does not accept a request body.

10.3.2.2. Response

Example 10.5. Validate token: JSON response

```
{
  "access": {
    "token": {
      "id": "ab48a9efdfedb23ty3494",
      "expires": "2010-11-01T03:32:15-05:00",
      "tenant": {
        "id": "345",
        "name": "My Project"
      }
    },
    "user": {
      "id": "123",
      "name": "jqsmith",
      "roles": [
        {
          "id": "234",
          "name": "compute:admin"
        }
      ]
    }
  }
}
```

```
        "id": "234",
        "name": "object-store:admin",
        "tenantId": "1"
      },
    ],
    "roles_links": []
  }
}
```


10.3.3. Validate token (admin)

Method	URI	Description
HEAD	/v2.0/tokens/{tokenId}?belongsTo	Validates a token and confirms that it belongs to a tenant, for performance.

Normal response codes: 200203204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.3.3.1. Request

This table shows the header parameters for the validate token (admin) request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the validate token (admin) request:

Name	Type	Description
{tokenId}	UUID	The authentication token for which to perform the operation.

This operation does not accept a request body.

10.3.4. Delete token

Method	URI	Description
DELETE	/v2.0/tokens/{tokenId}{?belongsTo}	Deletes a token.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.3.4.1. Request

This table shows the header parameters for the delete token request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the delete token request:

Name	Type	Description
{tokenId}	UUID	The authentication token for which to perform the operation.

This operation does not accept a request body.

10.3.5. List endpoints for token

Method	URI	Description
GET	/v2.0/tokens/{tokenId}/endpoints{?limit,marker}	Lists the endpoints associated with a token.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.3.5.1. Request

This table shows the header parameters for the list endpoints for token request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the list endpoints for token request:

Name	Type	Description
{tokenId}	UUID	The authentication token for which to perform the operation.

This operation does not accept a request body.

10.3.5.2. Response

Example 10.6. List endpoints for token: JSON response

```
{
  "endpoints_links": [],
  "endpoints": [
    {
      "name": "nova",
      "adminURL": "https://nova.region-one.internal.com/v2/be1319401cfa4a0aa590b97cc7b64d8d",
      "region": "RegionOne",
      "internalURL": "https://nova.region-one.internal.com/v2/be1319401cfa4a0aa590b97cc7b64d8d",
      "type": "compute",
      "id": "11b41ee1b00841128b7333d4bf1a6140",
      "publicURL": "https://nova.region-one.public.com/v2/be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
      "name": "neutron",
      "adminURL": "https://neutron.region-one.internal.com/",
      "region": "RegionOne",
      "internalURL": "https://neutron.region-one.internal.com/",
      "type": "network",
      "id": "cdbfa3c416d741a9b5c968f2dc628acb",
      "publicURL": "https://neutron.region-one.public.com/"
    }
  ]
}
```

```

        "name": "cinderv2",
        "adminURL": "https://cinderv2.region-one.internal.com/v2/
be1319401cfa4a0aa590b97cc7b64d8d",
        "region": "RegionOne",
        "internalURL": "https://cinderv2.region-one.internal.com/v2/
be1319401cfa4a0aa590b97cc7b64d8d",
        "type": "cinderv2v2",
        "id": "6de282e4132747ecb48f6fd8c525c6f6",
        "publicURL": "https://cinderv2.region-one.public.com/v2/
be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
        "name": "trove",
        "adminURL": "https://trove.region-one.internal.com/v1.0/
be1319401cfa4a0aa590b97cc7b64d8d",
        "region": "RegionOne",
        "internalURL": "https://trove.region-one.internal.com/v1.0/
be1319401cfa4a0aa590b97cc7b64d8d",
        "type": "database",
        "id": "4bfad53a0c684bd981d093099eb7799b",
        "publicURL": "https://trove.region-one.public.com/v1.0/
be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
        "name": "s3",
        "adminURL": "https://s3.region-one.internal.com",
        "region": "RegionOne",
        "internalURL": "https://s3.region-one.internal.com",
        "type": "s3",
        "id": "50fb6b43cde44bb6b0e471a682dc42dd",
        "publicURL": "https://s3.region-one.public.com"
    },
    {
        "name": "glance",
        "adminURL": "https://glance.region-one.internal.com",
        "region": "RegionOne",
        "internalURL": "https://glance.region-one.internal.com",
        "type": "glance",
        "id": "838a338171164c3c8f56e6b5882267ff",
        "publicURL": "https://glance.region-one.public.com"
    },
    {
        "name": "novav3",
        "adminURL": "https://novav3.region-one.internal.com/v3",
        "region": "RegionOne",
        "internalURL": "https://novav3.region-one.internal.com/v3",
        "type": "compute",
        "id": "b437edd03d244bf4be605b9b8c8689e0",
        "publicURL": "https://novav3.region-one.public.com/v3"
    },
    {
        "name": "heat",
        "adminURL": "https://heat.region-one.internal.com/v1",
        "region": "RegionOne",
        "internalURL": "https://heat.region-one.internal.com/v1",
        "type": "cloudformation",
        "id": "7a0f6f37344d488fa596a1325e0fcf10",
        "publicURL": "https://heat.region-one.public.com/v1"
    }
]

```

```

        "name": "cinder",
        "adminURL": "https://cinder.region-one.internal.com/v1/
be1319401cfa4a0aa590b97cc7b64d8d",
        "region": "RegionOne",
        "internalURL": "https://cinder.region-one.internal.com/v1/
be1319401cfa4a0aa590b97cc7b64d8d",
        "type": "cinderv2",
        "id": "d4f251065dce4ce3946d3c1b87e167f2",
        "publicURL": "https://cinder.region-one.public.com/v1/
be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
        "name": "ec2",
        "adminURL": "https://ec2.region-one.internal.com/services/Admin",
        "region": "RegionOne",
        "internalURL": "https://ec2.region-one.internal.com/services/
Cloud",
        "type": "ec2",
        "id": "44c6bf28d9bd4d63bfb00d66f22439a8",
        "publicURL": "https://ec2.region-one.public.com/services/Cloud"
    },
    {
        "name": "heat",
        "adminURL": "https://heat.region-one.internal.com/v1/
be1319401cfa4a0aa590b97cc7b64d8d",
        "region": "RegionOne",
        "internalURL": "https://heat.region-one.internal.com/v1/
be1319401cfa4a0aa590b97cc7b64d8d",
        "type": "orchestration",
        "id": "21aaace3f33c46b8aaea2b17d41ffd54",
        "publicURL": "https://heat.region-one.public.com/v1/
be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
        "name": "swift",
        "adminURL": "https://swift.region-one.internal.com",
        "region": "RegionOne",
        "internalURL": "https://swift.region-one.internal.com/v1/
AUTH_be1319401cfa4a0aa590b97cc7b64d8d",
        "type": "object-store",
        "id": "35f7aca3be384580a3b1df43a97c2eb2",
        "publicURL": "https://swift.region-one.public.com/v1/
AUTH_be1319401cfa4a0aa590b97cc7b64d8d"
    },
    {
        "name": "keystone",
        "adminURL": "https://keystone.region-one.internal.com/v2.0",
        "region": "RegionOne",
        "internalURL": "https://keystone.region-one.internal.com/v2.0",
        "type": "identity",
        "id": "48da758fb58c47dcaf02000a4409a265",
        "publicURL": "https://keystone.region-one.public.com/v2.0"
    }
]
}

```

10.4. Users

Method	URI	Description
GET	/v2.0/users{?name}	Lists all users.
POST	/v2.0/users	Creates a user.
PUT	/v2.0/users/{userId}	Updates a user.
DELETE	/v2.0/users/{userId}	Deletes a user.
GET	/v2.0/users/{userId}	Shows details for a user, by ID.
GET	/v2.0/users/{userId}/roles{?limit,marker}	Lists global roles for a user. Excludes tenant roles.

10.4.1. List users

Method	URI	Description
GET	/v2.0/users{?name}	Lists all users.

To show detailed information about a user by name, include the name query parameter in the request.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.4.1.1. Request

This table shows the query parameters for the list users request:

Name	Type	Description
name	String (Optional)	The user name. Specify the name query parameter as GET /v2.0/users?name={name}.

10.4.1.2. Response

Example 10.7. List users: JSON response

```
{
  "users": [
    {
      "username": "admin",
      "name": "admin",
      "enabled": true,
      "email": null,
      "id": "19dec86542d54bc791288b83d05c57a6"
    },
    {
      "username": "swift",
      "name": "swift",
      "enabled": true,
      "email": null,
      "id": "2109a7e134244071ac5b6ce31d8fe5b6"
    },
    {
      "username": "swiftusertest1",
      "name": "swiftusertest1",
      "enabled": true,
      "email": "test@example.com",
      "id": "3b59634090f84745bcd24bc28e564aff"
    },
    {
      "username": "alt_demo",
      "name": "alt_demo",
      "enabled": true,
      "email": "alt_demo@example.com",
      "id": "3f957317491c478daaee50992e5d2d3b"
    }
  ]
}
```

```

    {
      "username": "nova",
      "name": "nova",
      "enabled": true,
      "email": null,
      "id": "405ecdef1a434c70bb1e441cd295245d"
    },
    {
      "username": "swiftusertest3",
      "name": "swiftusertest3",
      "enabled": true,
      "email": "test3@example.com",
      "id": "5f58db25affc44c28d678279981c946f"
    },
    {
      "username": "swiftusertest2",
      "name": "swiftusertest2",
      "enabled": true,
      "email": "test2@example.com",
      "id": "9e4a840d8f5a45cdaa589febffcedb01"
    },
    {
      "username": "glance",
      "name": "glance",
      "enabled": true,
      "email": null,
      "id": "9f7df42d30264a0eb5f4e0d01486260d"
    },
    {
      "username": "demo",
      "name": "demo",
      "enabled": true,
      "email": "demo@example.com",
      "id": "a27a3939ad964215ad60315e8b2a3791"
    },
    {
      "username": "glance-swift",
      "name": "glance-swift",
      "enabled": true,
      "email": "glance-swift@example.com",
      "id": "bbe7ee42ffc345c18430599f74af9fa3"
    },
    {
      "username": "neutron",
      "name": "neutron",
      "enabled": true,
      "email": null,
      "id": "cf16ee300c2c412f81474ae452eda38d"
    },
    {
      "username": "cinder",
      "name": "cinder",
      "enabled": true,
      "email": null,
      "id": "f570fcf1692241978015cdb49242e383"
    }
  ]
}

```

```
{
```



```
"user": {  
  "username": "nova",  
  "name": "nova",  
  "enabled": true,  
  "email": null,  
  "id": "405ecdef1a434c70bb1e441cd295245d"  
}
```

10.4.2. Create user

Method	URI	Description
POST	/v2.0/users	Creates a user.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

10.4.2.1. Request

This table shows the header parameters for the create user request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

Example 10.8. Create user: JSON request

```
{
  "user": {
    "email": "new-user@example.com",
    "password": null,
    "enabled": true,
    "name": "new-user",
    "tenantId": "40429f980fac419bbfec372a5607c154"
  }
}
```

10.4.2.2. Response

Example 10.9. Create user: JSON response

```
{
  "user": {
    "username": "new-user",
    "name": "new-user",
    "id": "71767c619a90479ab21626abf76aa46c",
    "enabled": true,
    "email": "new-user@example.com",
    "tenantId": "40429f980fac419bbfec372a5607c154"
  }
}
```

10.4.3. Update user

Method	URI	Description
PUT	/v2.0/users/{userId}	Updates a user.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

10.4.3.1. Request

This table shows the URI parameters for the update user request:

Name	Type	Description
{userId}	String	The ID of the user for which you want to perform the request.

Example 10.10. Update user: JSON request

```
{
  "user": {
    "username": "nova",
    "name": "nova",
    "enabled": false,
    "email": null
  }
}
```

10.4.3.2. Response

Example 10.11. Update user: JSON response

```
{
  "user": {
    "username": "nova",
    "name": "nova",
    "enabled": false,
    "email": null,
    "id": "71767c619a90479ab21626abf76aa46c"
  }
}
```

10.4.4. Delete user

Method	URI	Description
DELETE	/v2.0/users/{userId}	Deletes a user.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.4.4.1. Request

This table shows the URI parameters for the delete user request:

Name	Type	Description
{userId}	String	The ID of the user for which you want to perform the request.

This operation does not accept a request body.

10.4.5. Show user details

Method	URI	Description
GET	/v2.0/users/{userId}	Shows details for a user, by ID.

The [openstack user show](#) command supports showing user details by name or ID. However, the command actually looks up the user ID for a user name and queries the user by ID.

As a workaround, complete these steps to show details for a user by name:

1. [List all users](#).
2. In the response, find the user name for which you want to show details and note its corresponding user ID.
3. [Show details for user](#).

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.4.5.1. Request

This table shows the URI parameters for the show user details request:

Name	Type	Description
{userId}	String	The ID of the user for which you want to perform the request.

This operation does not accept a request body.

10.4.5.2. Response

Example 10.12. Show user details: JSON response

```
{
  "user": {
    "username": "nova",
    "name": "nova",
    "enabled": true,
    "email": null,
    "id": "405ecdef1a434c70bb1e441cd295245d"
  }
}
```

10.4.6. List user global roles

Method	URI	Description
GET	/v2.0/users/{userId}/roles{?limit,marker}	Lists global roles for a user. Excludes tenant roles.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.4.6.1. Request

This table shows the URI parameters for the list user global roles request:

Name	Type	Description
{userId}	String	The ID of the user for which you want to perform the request.

This table shows the query parameters for the list user global roles request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

10.4.6.2. Response

Example 10.13. List user global roles: JSON response

```
{
  "roles": [
    {
      "id": "123",
      "name": "compute:admin",
      "description": "Nova Administrator"
    }
  ],
  "roles_links": []
}
```

10.5. Tenants

Method	URI	Description
GET	/v2.0/tenants{?limit,marker}	Lists all tenants.
GET	/v2.0/tenants{?name}	Shows details for a tenant, by name.
GET	/v2.0/tenants/{tenantId}	Shows details for a tenant, by ID.

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/users/{userId}/roles{?limit,marker}	Lists roles for a user on a tenant. Excludes global roles.

10.5.1. List tenants

Method	URI	Description
GET	/v2.0/tenants{?limit,marker}	Lists all tenants.

```
GET /v2.0/tenants HTTP/1.1
Host: identity.api.openstack.org
Content-Type: application/json
X-Auth-Token: fa8426a0-8eaf-4d22-8e13-7c1b16a9370c
Accept: application/json
```

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.5.1.1. Request

This table shows the header parameters for the list tenants request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the query parameters for the list tenants request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

10.5.1.2. Response

Example 10.14. Get tenants: JSON response

```
{
  "tenants": [
    {
      "id": "1234",
      "name": "ACME Corp",
      "description": "A description ...",
      "enabled": true
    },
    {
      "id": "3456",
      "name": "Iron Works",
      "description": "A description ...",
      "enabled": true
    }
  ]
}
```



```
    }  
  ],  
  "tenants_links": []  
}
```

10.5.2. Show tenant details, by name

Method	URI	Description
GET	/v2.0/tenants{?name}	Shows details for a tenant, by name.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.5.2.1. Request

This table shows the header parameters for the show tenant details, by name request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the query parameters for the show tenant details, by name request:

Name	Type	Description
name	String (Required)	Filters the response by a tenant name.

This operation does not accept a request body.

10.5.2.2. Response

Example 10.15. Show tenant by name: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

10.5.3. Show tenant details, by ID

Method	URI	Description
GET	/v2.0/tenants/{tenantId}	Shows details for a tenant, by ID.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.5.3.1. Request

This table shows the header parameters for the show tenant details, by id request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the show tenant details, by id request:

Name	Type	Description
{tenantId}	String	The tenant ID.

This operation does not accept a request body.

10.5.3.2. Response

Example 10.16. Show tenant by ID: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

10.5.4. List roles for user

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/users/{userId}/roles{?limit,marker}	Lists roles for a user on a tenant. Excludes global roles.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

10.5.4.1. Request

This table shows the header parameters for the list roles for user request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the list roles for user request:

Name	Type	Description
{tenantId}	String	The tenant ID.
{userId}	String	The user ID.

This table shows the query parameters for the list roles for user request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

10.5.4.2. Response

Example 10.17. List roles for user: JSON response

```
{
  "roles": [
    {
      "id": "123",
      "name": "compute:admin",
      "description": "Nova Administrator"
    }
  ],
  "roles_links": []
}
```

11. Identity API v2.0 extensions (SUPPORTED)

To list available Identity API v2.0 extensions, issue a **GET** request to `v2.0/extensions`.

Method	URI	Description
OS-KSADM admin extension		
GET	<code>/v2.0/users{?limit,marker}</code>	Lists all users.
POST	<code>/v2.0/users</code>	Creates a user.
PUT	<code>/v2.0/users/{userId}</code>	Updates a user.
DELETE	<code>/v2.0/users/{userId}</code>	Deletes a user.
PUT	<code>/v2.0/users/{userId}/OS-KSADM/enabled</code>	Enables a user.
GET	<code>/v2.0/users/{userId}/roles{?serviceId,limit,marker}</code>	Lists global roles for a user.
PUT	<code>/v2.0/users/{userId}/roles/OS-KSADM/{roleId}</code>	Grants a global role to a user.
DELETE	<code>/v2.0/users/{userId}/roles/OS-KSADM/{roleId}</code>	Deletes a global role from a user.
POST	<code>/v2.0/tenants</code>	Creates a tenant.
POST	<code>/v2.0/tenants/{tenantId}</code>	Updates a tenant.
DELETE	<code>/v2.0/tenants/{tenantId}</code>	Deletes a tenant.
GET	<code>/v2.0/tenants/{tenantId}/users{?limit,marker}</code>	Lists all users for a tenant.
PUT	<code>/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}</code>	Grants a role to a user for a tenant.
DELETE	<code>/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}</code>	Revokes a role from a user for a tenant.
POST	<code>/v2.0/OS-KSADM</code>	Creates a role.
GET	<code>/v2.0/OS-KSADM/roles/{role_name}</code>	Shows information for a role, by name.
GET	<code>/v2.0/OS-KSADM/{roleId}</code>	Shows details for a role, by ID.
DELETE	<code>/v2.0/OS-KSADM/{roleId}</code>	Deletes a role.
GET	<code>/v2.0/OS-KSADM/{?limit,marker}</code>	Lists all roles.
GET	<code>/v2.0/services{?limit,marker}</code>	Lists all services.
POST	<code>/v2.0/services</code>	Creates a service.
GET	<code>/v2.0/services/{serviceName}</code>	Shows information for a service, by name.
GET	<code>/v2.0/services/{serviceId}</code>	Shows information for a service, by ID.
DELETE	<code>/v2.0/services/{serviceId}</code>	Deletes a service.
OS-KSCATALOG admin extension		
GET	<code>/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints{?limit,marker}</code>	Lists endpoints for a tenant.
GET	<code>/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints</code>	Gets endpoint for a tenant.
POST	<code>/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints</code>	Creates endpoint to a tenant.
DELETE	<code>/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints/{endpointId}</code>	Deletes an endpoint from a tenant.

Method	URI	Description
GET	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId, limit, marker}	Lists endpoint templates.
GET	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Gets endpoint templates.
POST	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Creates endpoint template.
PUT	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Updates endpoint template.
DELETE	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Deletes an endpoint template.
OS-KSEC2 admin extension		
GET	/v2.0/users/{userId}/OS-KSADM/credentials{?limit, marker}	Lists credentials.
POST	/v2.0/users/{userId}/OS-KSADM/credentials	Grants a credential to a user.
GET	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Gets user credentials.
POST	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Updates credentials for a user.
DELETE	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Deletes user credentials.
GET	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials/{type}{?type, limit, marker}	Lists credentials by type.
OS-KSS3 admin extension		
GET	/v2.0/users/{userId}/OS-KSS3/credentials{?limit, marker}	Lists credentials.
POST	/v2.0/users/{userId}/OS-KSS3/credentials	Grants a credential to a user.
GET	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials	Gets user credentials.
POST	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials	Updates credentials.
DELETE	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials	Revokes user credentials.
GET	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials/{type}{?limit, marker, type}	Lists credentials by type.
OS-KSVALIDATE admin extension		
GET	/v2.0/OS-KSVALIDATE/token/validate{?belongsTo, HP-IDM-serviceId}	Checks that a token is valid and that it belongs to the tenant and any service IDs. Returns the permissions for a particular client.
HEAD	/v2.0/OS-KSVALIDATE/token/validate{?belongsTo, HP-IDM-serviceId}	Checks that a token is valid and that it belongs to the tenant and any service IDs, for performance.
GET	/v2.0/OS-KSVALIDATE/token/endpoints{?HP-IDM-serviceId, limit, marker}	Lists endpoints for a token.

11.1. OS-KSADM admin extension

Supports create, read, update, and delete (CRUD) operations for users, tenants, roles, and services. Requires administrator privileges.

Method	URI	Description
GET	/v2.0/users{?limit,marker}	Lists all users.
POST	/v2.0/users	Creates a user.
PUT	/v2.0/users/{userId}	Updates a user.
DELETE	/v2.0/users/{userId}	Deletes a user.
PUT	/v2.0/users/{userId}/OS-KSADM/enabled	Enables a user.
GET	/v2.0/users/{userId}/roles{?serviceId,limit,marker}	Lists global roles for a user.
PUT	/v2.0/users/{userId}/roles/OS-KSADM/{roleId}	Grants a global role to a user.
DELETE	/v2.0/users/{userId}/roles/OS-KSADM/{roleId}	Deletes a global role from a user.
POST	/v2.0/tenants	Creates a tenant.
POST	/v2.0/tenants/{tenantId}	Updates a tenant.
DELETE	/v2.0/tenants/{tenantId}	Deletes a tenant.
GET	/v2.0/tenants/{tenantId}/users{?limit,marker}	Lists all users for a tenant.
PUT	/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}	Grants a role to a user for a tenant.
DELETE	/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}	Revokes a role from a user for a tenant.
POST	/v2.0/OS-KSADM	Creates a role.
GET	/v2.0/OS-KSADM/roles/{role_name}	Shows information for a role, by name.
GET	/v2.0/OS-KSADM/{roleId}	Shows details for a role, by ID.
DELETE	/v2.0/OS-KSADM/{roleId}	Deletes a role.
GET	/v2.0/OS-KSADM/{?limit,marker}	Lists all roles.
GET	/v2.0/services{?limit,marker}	Lists all services.
POST	/v2.0/services	Creates a service.
GET	/v2.0/services/{serviceName}	Shows information for a service, by name.
GET	/v2.0/services/{serviceId}	Shows information for a service, by ID.
DELETE	/v2.0/services/{serviceId}	Deletes a service.

11.1.1. List users

Method	URI	Description
GET	/v2.0/users{?limit,marker}	Lists all users.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.1.1.1. Request

This table shows the header parameters for the list users request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the query parameters for the list users request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

11.1.1.2. Response

Example 11.1. List users: JSON response

```
{
  "users": [
    {
      "id": "u1000",
      "name": "jqsmith",
      "email": "john.smith@example.org",
      "enabled": true
    },
    {
      "id": "u1001",
      "name": "jqsmith",
      "email": "john.smith@example.org",
      "enabled": true
    }
  ],
  "users_links": []
}
```


11.1.2. Create user

Method	URI	Description
POST	/v2.0/users	Creates a user.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.2.1. Request

This table shows the header parameters for the create user request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

Example 11.2. Create user: JSON request

```
{
  "user": {
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true,
    "OS-KSADM:password": "secretsecret"
  }
}
```

11.1.2.2. Response

Example 11.3. Create user: JSON response

```
{
  "user": {
    "id": "u1000",
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

11.1.3. Update user

Method	URI	Description
PUT	/v2.0/users/{userId}	Updates a user.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.3.1. Request

This table shows the header parameters for the update user request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the update user request:

Name	Type	Description
{userId}	String	The ID of the user for which to perform the request.

Example 11.4. Update user: JSON request

```
{
  "user": {
    "id": "u1000",
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

11.1.3.2. Response

Example 11.5. Update user: JSON response

```
{
  "user": {
    "id": "u1000",
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

11.1.4. Delete user

Method	URI	Description
DELETE	/v2.0/users/{userId}	Deletes a user.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.4.1. Request

This table shows the header parameters for the delete user request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the delete user request:

Name	Type	Description
{userId}	String	The ID of the user for which to perform the request.

This operation does not accept a request body.

11.1.5. Enable user

Method	URI	Description
PUT	/v2.0/users/{userId}/OS-KSADM/enabled	Enables a user.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.5.1. Request

This table shows the header parameters for the enable user request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the enable user request:

Name	Type	Description
{userId}	String	The ID of the user for which to perform the request.

Example 11.6. Enable user: JSON request

```
{
  "user": {
    "enabled": true
  }
}
```

11.1.5.2. Response

Example 11.7. Enable user: JSON response

```
{
  "user": {
    "id": "u1000",
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

11.1.6. List global roles for user

Method	URI	Description
GET	/v2.0/users/{userId}/roles{?serviceId,limit,marker}	Lists global roles for a user.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.1.6.1. Request

This table shows the header parameters for the list global roles for user request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the list global roles for user request:

Name	Type	Description
{userId}	String	The ID of the user for which to perform the request.

This table shows the query parameters for the list global roles for user request:

Name	Type	Description
serviceId	String (Optional)	The service ID.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

11.1.6.2. Response

Example 11.8. List global roles for user: JSON response

```
{
  "roles": [
    {
      "id": "8341d3603a1d4d5985bff09f10704d4d",
      "name": "service"
    },
    {
      "id": "2e66d57df76946fdb034bc4da6fdec0",
      "name": "admin"
    }
  ]
}
```

```
}  
    ]  
}
```

11.1.7. Grant global role to user

Method	URI	Description
PUT	/v2.0/users/{userId}/roles/OS-KSADM/{roleId}	Grants a global role to a user.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.7.1. Request

This table shows the header parameters for the grant global role to user request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the grant global role to user request:

Name	Type	Description
{userId}	String	The ID of the user for which to perform the request.
{roleId}	Int	The role ID.

This operation does not accept a request body.

11.1.8. Delete global role from user

Method	URI	Description
DELETE	/v2.0/users/{userId}/roles/OS-KSADM/{roleId}	Deletes a global role from a user.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.8.1. Request

This table shows the header parameters for the delete global role from user request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the delete global role from user request:

Name	Type	Description
{userId}	String	The ID of the user for which to perform the request.
{roleId}	Int	The role ID.

This operation does not accept a request body.

11.1.9. Create tenant

Method	URI	Description
POST	/v2.0/tenants	Creates a tenant.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.9.1. Request

This table shows the header parameters for the create tenant request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

Example 11.9. Create tenant: JSON request

```
{
  "tenant": {
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

11.1.9.2. Response

Example 11.10. Create tenant: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

11.1.10. Update tenant

Method	URI	Description
POST	/v2.0/tenants/{tenantId}	Updates a tenant.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.10.1. Request

This table shows the header parameters for the update tenant request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the update tenant request:

Name	Type	Description
{tenantId}	String	The tenant ID.

Example 11.11. Update tenant: JSON request

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

11.1.10.2. Response

Example 11.12. Update tenant: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

11.1.11. Delete tenant

Method	URI	Description
DELETE	/v2.0/tenants/{tenantId}	Deletes a tenant.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.11.1. Request

This table shows the header parameters for the delete tenant request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the delete tenant request:

Name	Type	Description
{tenantId}	String	The tenant ID.

This operation does not accept a request body.

11.1.12. List users on a tenant

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/users{?limit,marker}	Lists all users for a tenant.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.1.12.1. Request

This table shows the header parameters for the list users on a tenant request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the list users on a tenant request:

Name	Type	Description
{tenantId}	String	The tenant ID.

This table shows the query parameters for the list users on a tenant request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

11.1.12.2. Response

Example 11.13. List users on a tenant: JSON response

```
{
  "users": [
    {
      "id": "u1000",
      "name": "jqsmith",
      "email": "john.smith@example.org",
      "enabled": true
    },
    {
      "id": "u1001",
      "name": "jqsmith",
      "email": "john.smith@example.org",
```

```
        "enabled": true
      }
    ],
    "users_links": []
  }
```

11.1.13. Grant roles to user on tenant

Method	URI	Description
PUT	/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}	Grants a role to a user for a tenant.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503), Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.1.13.1. Request

This table shows the header parameters for the grant roles to user on tenant request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the grant roles to user on tenant request:

Name	Type	Description
{tenantId}	String	The tenant ID.
{userId}	String	The user ID.
{roleId}	String	The role ID.

This operation does not accept a request body.

11.1.14. Revoke role from user on tenant

Method	URI	Description
DELETE	/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}	Revokes a role from a user for a tenant.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.14.1. Request

This table shows the header parameters for the revoke role from user on tenant request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the revoke role from user on tenant request:

Name	Type	Description
{tenantId}	String	The tenant ID.
{userId}	String	The user ID.
{roleId}	String	The role ID.

This operation does not accept a request body.

11.1.15. Create role

Method	URI	Description
POST	/v2.0/OS-KSADM	Creates a role.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.15.1. Request

This table shows the header parameters for the create role request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

Example 11.14. Create role: JSON request

```
{
  "role": {
    "id": "123",
    "name": "Guest",
    "description": "Guest Access"
  }
}
```

11.1.15.2. Response

Example 11.15. Create role: JSON response

```
{
  "role": {
    "id": "123",
    "name": "Guest",
    "description": "Guest Access"
  }
}
```


11.1.16. Show role information by name

Method	URI	Description
GET	/v2.0/OS-KSADM/roles/{role_name}	Shows information for a role, by name.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.16.1. Request

This table shows the header parameters for the show role information by name request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the show role information by name request:

Name	Type	Description
{name}	String	The role name.

This operation does not accept a request body.

11.1.16.2. Response

Example 11.16. Show role information by name: JSON response

```
{
  "role": {
    "id": "123",
    "name": "Guest",
    "description": "Guest Access"
  }
}
```

11.1.17. Show role details, by ID

Method	URI	Description
GET	/v2.0/OS-KSADM/{roleId}	Shows details for a role, by ID.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.17.1. Request

This table shows the header parameters for the show role details, by id request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the show role details, by id request:

Name	Type	Description
{roleId}	String	The role ID.

This operation does not accept a request body.

11.1.17.2. Response

Example 11.17. Show role details, by ID: JSON response

```
{
  "role": {
    "id": "123",
    "name": "Guest",
    "description": "Guest Access"
  }
}
```

11.1.18. Delete role

Method	URI	Description
DELETE	/v2.0/OS-KSADM/{roleId}	Deletes a role.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.18.1. Request

This table shows the header parameters for the delete role request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the delete role request:

Name	Type	Description
{roleId}	String	The role ID.

This operation does not accept a request body.

11.1.19. List roles

Method	URI	Description
GET	/v2.0/OS-KSADM/{?limit,marker}	Lists all roles.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.1.19.1. Request

This table shows the header parameters for the list roles request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the query parameters for the list roles request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

11.1.19.2. Response

Example 11.18. List roles: JSON response

```
{
  "roles": [
    {
      "id": "123",
      "name": "compute:admin",
      "description": "Nova Administrator"
    }
  ],
  "roles_links": []
}
```

11.1.20. List services

Method	URI	Description
GET	/v2.0/services{?limit,marker}	Lists all services.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.1.20.1. Request

This table shows the header parameters for the list services request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the query parameters for the list services request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

11.1.20.2. Response

Example 11.19. List services: JSON response

```
{
  "OS-KSADM:services": [
    {
      "id": "123",
      "name": "nova",
      "type": "compute",
      "description": "OpenStack Compute Service"
    },
    {
      "id": "234",
      "name": "glance",
      "type": "image",
      "description": "OpenStack Image Service"
    }
  ],
  "OS-KSADM:services_links": []
}
```

Example 11.20. List services: JSON response

```
{
  "OS-KSADM:services": [
    {
      "id": "123",
      "name": "nova",
      "type": "compute",
      "description": "OpenStack Compute Service"
    },
    {
      "id": "234",
      "name": "glance",
      "type": "image",
      "description": "OpenStack Image Service"
    }
  ],
  "OS-KSADM:services_links": []
}
```

11.1.21. Create service

Method	URI	Description
POST	/v2.0/services	Creates a service.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.21.1. Request

This table shows the header parameters for the create service request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

Example 11.21. Create service: JSON request

```
{
  "OS-KSADM:service": {
    "id": "123",
    "name": "nova",
    "type": "compute",
    "description": "OpenStack Compute Service"
  }
}
```

11.1.21.2. Response

Example 11.22. Create service: JSON response

```
{
  "OS-KSADM:service": {
    "id": "123",
    "name": "nova",
    "type": "compute",
    "description": "OpenStack Compute Service"
  }
}
```

11.1.22. Show service information by name

Method	URI	Description
GET	/v2.0/services/{serviceName}	Shows information for a service, by name.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.1.22.1. Request

This table shows the header parameters for the show service information by name request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the show service information by name request:

Name	Type	Description
{name}	String	The service name.

This operation does not accept a request body.

11.1.22.2. Response

Example 11.23. Show service information by name: JSON response

```
{
  "OS-KSADM:service": {
    "id": "123",
    "name": "nova",
    "type": "compute",
    "description": "OpenStack Compute Service"
  }
}
```


11.1.23. Shows service information by ID

Method	URI	Description
GET	/v2.0/services/{serviceId}	Shows information for a service, by ID.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.1.23.1. Request

This table shows the header parameters for the shows service information by id request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the shows service information by id request:

Name	Type	Description
{serviceId}	String	The service ID.

This operation does not accept a request body.

11.1.23.2. Response

Example 11.24. Shows service information by ID: JSON response

```
{
  "OS-KSADM:service": {
    "id": "123",
    "name": "nova",
    "type": "compute",
    "description": "OpenStack Compute Service"
  }
}
```

11.1.24. Delete service

Method	URI	Description
DELETE	/v2.0/services/{serviceId}	Deletes a service.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.1.24.1. Request

This table shows the header parameters for the delete service request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the delete service request:

Name	Type	Description
{serviceId}	String	The service ID.

This operation does not accept a request body.

11.2. OS-KSCATALOG admin extension

Supports CRUD operations for endpoint templates and endpoints. Requires administrator privileges.

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints{?limit,marker}	Lists endpoints for a tenant.
GET	/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints	Gets endpoint for a tenant.
POST	/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints	Creates endpoint to a tenant.
DELETE	/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints/{endpointId}	Deletes an endpoint from a tenant.
GET	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId,limit,marker}	Lists endpoint templates.
GET	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Gets endpoint templates.
POST	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Creates endpoint template.
PUT	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Updates endpoint template.
DELETE	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Deletes an endpoint template.

11.2.1. List endpoints

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints{?limit,marker}	Lists endpoints for a tenant.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.2.1.1. Request

This table shows the header parameters for the list endpoints request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the list endpoints request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list endpoints request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

11.2.1.2. Response

Example 11.25. List endpoints: JSON response

```
{
  "endpoints": [
    {
      "id": 1,
      "tenantId": "1",
      "region": "North",
      "type": "compute",
      "publicURL": "https://compute.north.public.com/v1",
      "internalURL": "https://compute.north.internal.com/v1",
      "adminURL": "https://compute.north.internal.com/v1"
    },
    {
```

```
        "id": 2,  
        "tenantId": "1",  
        "region": "South",  
        "type": "compute",  
        "publicURL": "https://compute.north.public.com/v1",  
        "internalURL": "https://compute.north.internal.com/v1",  
        "adminURL": "https://compute.north.internal.com/v1"  
    },  
    {  
        "id": 3,  
        "tenantId": "1",  
        "region": "East",  
        "type": "compute",  
        "publicURL": "https://compute.north.public.com/v1",  
        "internalURL": "https://compute.north.internal.com/v1",  
        "adminURL": "https://compute.north.internal.com/v1"  
    },  
    {  
        "id": 4,  
        "tenantId": "1",  
        "region": "West",  
        "type": "compute",  
        "publicURL": "https://compute.north.public.com/v1",  
        "internalURL": "https://compute.north.internal.com/v1",  
        "adminURL": "https://compute.north.internal.com/v1"  
    },  
    {  
        "id": 5,  
        "tenantId": "1",  
        "region": "Global",  
        "type": "compute",  
        "publicURL": "https://compute.north.public.com/v1",  
        "internalURL": "https://compute.north.internal.com/v1",  
        "adminURL": "https://compute.north.internal.com/v1"  
    }  
],  
"endpoints_links": []  
}
```

11.2.2. Get endpoint

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints	Gets endpoint for a tenant.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.2.2.1. Request

This table shows the header parameters for the get endpoint request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the get endpoint request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

11.2.2.2. Response

Example 11.26. Get endpoint: JSON response

```
{
  "endpoint": {
    "id": 1,
    "tenantId": 1,
    "region": "North",
    "type": "compute",
    "publicURL": "https://compute.north.public.com/v1",
    "internalURL": "https://compute.north.internal.com/v1",
    "adminURL": "https://compute.north.internal.com/v1"
  }
}
```

11.2.3. Create endpoint

Method	URI	Description
POST	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints	Creates endpoint to a tenant.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.2.3.1. Request

This table shows the header parameters for the create endpoint request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the create endpoint request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 11.27. Create endpoint: JSON request

```
{
  "OS-KSCATALOG:endpointTemplate": {
    "id": 1
  }
}
```

11.2.3.2. Response

Example 11.28. Create endpoint: JSON response

```
{
  "endpoint": {
    "id": 1,
    "tenantId": 1,
    "region": "North",
    "type": "compute",
    "publicURL": "https://compute.north.public.com/v1",
    "internalURL": "https://compute.north.internal.com/v1",
    "adminURL": "https://compute.north.internal.com/v1"
  }
}
```

11.2.4. Delete endpoint

Method	URI	Description
DELETE	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints/{endpointId}	Deletes an endpoint from a tenant.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.2.4.1. Request

This table shows the header parameters for the delete endpoint request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the delete endpoint request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{endpoint_id}	UUID	The UUID of the endpoint.

This operation does not accept a request body.

11.2.5. List endpoint templates

Method	URI	Description
GET	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId,limit,marker}	Lists endpoint templates.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.2.5.1. Request

This table shows the header parameters for the list endpoint templates request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the query parameters for the list endpoint templates request:

Name	Type	Description
serviceId	String (Optional)	The service ID.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

11.2.5.2. Response

Example 11.29. List endpoint templates: JSON response

```
{
  "OS-KSCATALOG:endpointsTemplates": [
    {
      "id": 1,
      "region": "North",
      "global": true,
      "type": "compute",
      "publicURL": "https://compute.north.public.com/v1",
      "internalURL": "https://compute.north.internal.com/v1",
      "enabled": true
    },
    {
      "id": 2,
      "region": "South",
      "type": "compute",
```



```
        "publicURL": "https://compute.south.public.com/v1",
        "internalURL": "https://compute.south.internal.com/v1",
        "enabled": false
    },
    {
        "id": 3,
        "region": "North",
        "global": true,
        "type": "object-store",
        "publicURL": "https://object-store.north.public.com/v1.0",
        "enabled": true
    },
    {
        "id": 4,
        "region": "South",
        "type": "object-store",
        "publicURL": "https://object-store.south.public.com/v2",
        "enabled": true
    },
    {
        "id": 5,
        "global": true,
        "type": "OS-DNS:DNS",
        "publicURL": "https://dns.public.com/v3.2",
        "enabled": true
    }
],
"OS-KSCATALOG:endpointsTemplates_links": []
}
```

11.2.6. Get endpoint template

Method	URI	Description
GET	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Gets endpoint templates.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.2.6.1. Request

This table shows the header parameters for the get endpoint template request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This operation does not accept a request body.

11.2.6.2. Response

Example 11.30. Get endpoint template: JSON response

```
{
  "OS-KSCATALOG:endpointTemplate": {
    "id": 1,
    "region": "North",
    "global": true,
    "type": "compute",
    "publicURL": "https://compute.north.public.com/v1",
    "internalURL": "https://compute.north.internal.com/v1",
    "enabled": true
  }
}
```

11.2.7. Create endpoint template

Method	URI	Description
POST	/v2.0/OS-KSCATALOG/endpointTemplates/{?serviceId}	Creates endpoint template.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.2.7.1. Request

This table shows the header parameters for the create endpoint template request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

Example 11.31. Create endpoint template: JSON request

```
{
  "OS-KSCATALOG:endpointTemplate": {
    "id": 1,
    "region": "North",
    "global": true,
    "type": "compute",
    "publicURL": "https://compute.north.public.com/v1",
    "internalURL": "https://compute.north.internal.com/v1",
    "versionId": "1",
    "versionInfo": "https://compute.north.public.com/v1/",
    "versionList": "https://compute.north.public.com/",
    "enabled": true
  }
}
```

11.2.7.2. Response

Example 11.32. Create endpoint template: JSON response

```
{
  "OS-KSCATALOG:endpointTemplate": {
    "id": 1,
    "region": "North",
    "global": true,
    "type": "compute",
    "publicURL": "https://compute.north.public.com/v1",
    "internalURL": "https://compute.north.internal.com/v1",
    "enabled": true
  }
}
```

11.2.8. Update endpoint template

Method	URI	Description
PUT	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Updates endpoint template.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.2.8.1. Request

This table shows the header parameters for the update endpoint template request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the update endpoint template request:

Name	Type	Description
{endpointTemplateId}	String	The endpoint template ID.

Example 11.33. Update endpoint template: JSON request

```
{
  "OS-KSCATALOG:endpointTemplate": {
    "id": 1,
    "region": "North",
    "global": true,
    "type": "compute",
    "publicURL": "https://compute.north.public.com/v1",
    "internalURL": "https://compute.north.internal.com/v1",
    "versionId": "1",
    "versionInfo": "https://compute.north.public.com/v1/",
    "versionList": "https://compute.north.public.com/",
    "enabled": true
  }
}
```

11.2.8.2. Response

Example 11.34. Update endpoint template: JSON response

```
{
  "OS-KSCATALOG:endpointTemplate": {
    "id": 1,
    "region": "North",
    "global": true,
    "type": "compute",
    "publicURL": "https://compute.north.public.com/v1",
```

```
        "internalURL": "https://compute.north.internal.com/v1",  
        "enabled": true  
    }  
}
```

11.2.9. Delete endpoint template.

Method	URI	Description
DELETE	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Deletes an endpoint template.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.2.9.1. Request

This table shows the header parameters for the delete endpoint template. request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the delete endpoint template. request:

Name	Type	Description
{endpointTemplateId}	String	The endpoint template ID.

This operation does not accept a request body.

11.3. OS-KSEC2 admin extension

Supports Amazon Elastic Compute (EC2) style authentication.

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSADM/credentials{?limit,marker}	Lists credentials.
POST	/v2.0/users/{userId}/OS-KSADM/credentials	Grants a credential to a user.
GET	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Gets user credentials.
POST	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Updates credentials for a user.
DELETE	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Deletes user credentials.
GET	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials/{type}{?type,limit,marker}	Lists credentials by type.

11.3.1. List credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSADM/credentials{?limit,marker}	Lists credentials.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.3.1.1. Request

This table shows the header parameters for the list credentials request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the list credentials request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

This table shows the query parameters for the list credentials request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

11.3.1.2. Response

Example 11.35. List credentials: JSON response

```
{
  "credentials": [
    {
      "passwordCredentials": {
        "username": "test_user",
        "password": "secretsecret"
      }
    },
    {
      "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "secret": "secretsecret",

```

```
        "signature": "bbb"
      }
    },
    "credentials_links": []
  }
```


11.3.2. Grant credential to user

Method	URI	Description
POST	/v2.0/users/{userId}/OS-KSADM/credentials	Grants a credential to a user.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.3.2.1. Request

This table shows the header parameters for the grant credential to user request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the grant credential to user request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

Example 11.36. Grant credential to user: JSON request

```
{
  "OS-KSEC2-ec2Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

11.3.2.2. Response

Example 11.37. Grant credential to user: JSON response

```
{
  "OS-KSEC2-ec2Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

11.3.3. Get user credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Gets user credentials.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.3.3.1. Request

This table shows the header parameters for the get user credentials request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the get user credentials request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

This operation does not accept a request body.

11.3.3.2. Response

Example 11.38. Get user credentials: JSON response

```
{
  "OS-KSEC2-ec2Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

11.3.4. Update user credentials

Method	URI	Description
POST	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Updates credentials for a user.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.3.4.1. Request

This table shows the header parameters for the update user credentials request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the update user credentials request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

Example 11.39. Update user credentials: JSON request

```
{
  "OS-KSEC2-ec2Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

11.3.4.2. Response

Example 11.40. Update user credentials: JSON response

```
{
  "OS-KSEC2-ec2Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

11.3.5. Delete user credentials

Method	URI	Description
DELETE	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Deletes user credentials.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.3.5.1. Request

This table shows the header parameters for the delete user credentials request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the delete user credentials request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

This operation does not accept a request body.

11.3.6. List credentials by type

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials/{type}{?type,limit,marker}	Lists credentials by type.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.3.6.1. Request

This table shows the header parameters for the list credentials by type request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the list credentials by type request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

This table shows the query parameters for the list credentials by type request:

Name	Type	Description
type	String (Required)	The credential type.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

11.3.6.2. Response

Example 11.41. List credentials by type: JSON response

```
{
  "credentials": [
    {
      "passwordCredentials": {
        "username": "test_user",
        "password": "secretsecret"
      }
    }
  ]
}
```

```
    ],  
    "credentials_links": []  
}
```

11.4. OS-KSS3 admin extension

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSS3/credentials{?limit,marker}	Lists credentials.
POST	/v2.0/users/{userId}/OS-KSS3/credentials	Grants a credential to a user.
GET	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials	Gets user credentials.
POST	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials	Updates credentials.
DELETE	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials	Revokes user credentials.
GET	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials/{type}{?limit,marker,type}	Lists credentials by type.

11.4.1. List credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSS3/credentials{?limit,marker}	Lists credentials.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.4.1.1. Request

This table shows the header parameters for the list credentials request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the list credentials request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

This table shows the query parameters for the list credentials request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

11.4.1.2. Response

Example 11.42. List credentials: JSON response

```
{
  "credentials": [
    {
      "passwordCredentials": {
        "username": "test_user",
        "password": "secretsecret"
      }
    },
    {
      "OS-KSS3:s3Credentials": {
        "username": "test_user",
        "secret": "secretsecret",
```

```
        "signature": "bbb"
      }
    },
    "credentials_links": []
  }
```


11.4.2. Grant credential to user

Method	URI	Description
POST	/v2.0/users/{userId}/OS-KSS3/credentials	Grants a credential to a user.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.4.2.1. Request

This table shows the header parameters for the grant credential to user request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the grant credential to user request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

Example 11.43. Grant credential to user: JSON request

```
{
  "OS-KSS3:s3Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

11.4.2.2. Response

Example 11.44. Grant credential to user: JSON response

```
{
  "OS-KSS3:s3Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

11.4.3. Get user credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials	Gets user credentials.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.4.3.1. Request

This table shows the header parameters for the get user credentials request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the get user credentials request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

This operation does not accept a request body.

11.4.3.2. Response

Example 11.45. Get user credentials: JSON response

```
{
  "OS-KSS3:s3Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

11.4.4. Update user credentials

Method	URI	Description
POST	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials	Updates credentials.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.4.4.1. Request

This table shows the header parameters for the update user credentials request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the update user credentials request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

Example 11.46. Update user credentials: JSON request

```
{
  "OS-KSS3:s3Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

11.4.4.2. Response

Example 11.47. Update user credentials: JSON response

```
{
  "OS-KSS3:s3Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

11.4.5. Revoke user credentials

Method	URI	Description
DELETE	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials	Revokes user credentials.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), TenantConflictFault (409), Request Entity Too Large (413), Unsupported Media Type (415), Service Unavailable (503)

11.4.5.1. Request

This table shows the header parameters for the revoke user credentials request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the revoke user credentials request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

This operation does not accept a request body.

11.4.6. List credentials by type

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials/{type}?limit,marker,type	Lists credentials by type.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.4.6.1. Request

This table shows the header parameters for the list credentials by type request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.

This table shows the URI parameters for the list credentials by type request:

Name	Type	Description
{user_id}	UUID	The UUID of the user.

This table shows the query parameters for the list credentials by type request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
type	String (Required)	The credential type.

11.4.6.2. Response

Example 11.48. List credentials by type: JSON response

```
{
  "credentials": [
    {
      "passwordCredentials": {
        "username": "test_user",
        "password": "secretsecret"
      }
    },
  ],
}
```

```
{
  "OS-KSS3:s3Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
},
"credentials_links": []
}
```

11.5. OS-KSVALIDATE admin extension

Method	URI	Description
GET	/v2.0/OS-KSVALIDATE/token/validate {?belongsTo,HP-IDM-serviceId}	Checks that a token is valid and that it belongs to the tenant and any service IDs. Returns the permissions for a particular client.
HEAD	/v2.0/OS-KSVALIDATE/token/validate {?belongsTo,HP-IDM-serviceId}	Checks that a token is valid and that it belongs to the tenant and any service IDs, for performance.
GET	/v2.0/OS-KSVALIDATE/token/end-points {?HP-IDM-serviceId, limit, marker}	Lists endpoints for a token.

11.5.1. Validate token

Method	URI	Description
GET	/v2.0/OS-KSVALIDATE/token/validate {?belongsTo,HP-IDM-serviceId}	Checks that a token is valid and that it belongs to the tenant and any service IDs. Returns the permissions for a particular client.

Behavior is similar to /tokens/{tokenId}. If the token is not valid, the call returns the `itemNotFound` (404) response code.

This extension might decrypt the X-Subject-Token header and internally call and pass in all headers and query parameters to the normal validation code for Identity. Consequently, this extension must support all existing /tokens/{tokenId} calls including extensions such as HP-IDM.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.5.1.1. Request

This table shows the header parameters for the validate token request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.
X-Subject-Token	String (Required)	A valid authentication token.

This operation does not accept a request body.

11.5.1.2. Response

Example 11.49. Validate token: JSON response

```
{
  "access": {
    "token": {
      "id": "ab48a9efdfedb23ty3494",
      "expires": "2010-11-01T03:32:15-05:00",
      "tenant": {
        "id": "345",
        "name": "My Project"
      }
    },
    "user": {
      "id": "123",
      "name": "jqsmith",
      "roles": [
        {
```

```
        "id": "234",
        "name": "compute:admin"
      },
      {
        "id": "234",
        "name": "object-store:admin",
        "tenantId": "1"
      }
    ],
    "roles_links": []
  }
}
```


11.5.2. Check token

Method	URI	Description
HEAD	/v2.0/OS-KSVALIDATE/token/validate {?belongsTo, HP-IDM-serviceId}	Checks that a token is valid and that it belongs to the tenant and any service IDs, for performance.

Behavior is similar to /tokens/{tokenId}. If the token is not valid, the call returns the `itemNotFound` (404) response code.

This extension might decrypt the X-Subject-Token header and internally call and pass in all headers and query parameters to the normal validation code for Identity. Consequently, this extension must support all existing /tokens/{tokenId} calls including extensions such as HP-IDM.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.5.2.1. Request

This table shows the header parameters for the check token request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.
X-Subject-Token	String (Required)	A valid authentication token.

This operation does not accept a request body.

11.5.3. List endpoints for token

Method	URI	Description
GET	/v2.0/OS-KSVALIDATE/token/end-points{?HP-IDM-serviceId, limit, marker}	Lists endpoints for a token.

Normal response codes: 200203

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Not Found (404), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

11.5.3.1. Request

This table shows the header parameters for the list endpoints for token request:

Name	Type	Description
X-Auth-Token	String (Required)	A valid authentication token for an administrative user.
X-Subject-Token	String (Required)	A valid authentication token.

This table shows the query parameters for the list endpoints for token request:

Name	Type	Description
HP-IDM-serviceId	String (Optional)	One or more service IDs. Specify multiple values as a comma-separated list. The API checks roles against the service IDs. If a service ID is not valid or no roles are associated with it, this call returns the unauthorized (401) response code.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

11.5.3.2. Response

Example 11.50. List endpoints for token: JSON response

```
{
  "endpoints": [
    {
      "id": 1,
      "tenantId": "1",
      "region": "North",
      "type": "compute",
      "publicURL": "https://compute.north.public.com/v1",
      "internalURL": "https://compute.north.internal.com/v1",
      "adminURL": "https://compute.north.internal.com/v1"
    }
  ]
}
```

```
    },
    {
      "id": 2,
      "tenantId": "1",
      "region": "South",
      "type": "compute",
      "publicURL": "https://compute.north.public.com/v1",
      "internalURL": "https://compute.north.internal.com/v1",
      "adminURL": "https://compute.north.internal.com/v1"
    },
    {
      "id": 3,
      "tenantId": "1",
      "region": "East",
      "type": "compute",
      "publicURL": "https://compute.north.public.com/v1",
      "internalURL": "https://compute.north.internal.com/v1",
      "adminURL": "https://compute.north.internal.com/v1"
    },
    {
      "id": 4,
      "tenantId": "1",
      "region": "West",
      "type": "compute",
      "publicURL": "https://compute.north.public.com/v1",
      "internalURL": "https://compute.north.internal.com/v1",
      "adminURL": "https://compute.north.internal.com/v1"
    },
    {
      "id": 5,
      "tenantId": "1",
      "region": "Global",
      "type": "compute",
      "publicURL": "https://compute.north.public.com/v1",
      "internalURL": "https://compute.north.internal.com/v1",
      "adminURL": "https://compute.north.internal.com/v1"
    }
  ],
  "endpoints_links": []
}
```

12. Image service API v2 (CURRENT)

Image service API v2.0, API v2.1, API v2.2, and API v2.3.

Cloud providers can configure property protections that prevent non-administrative users from updating and deleting protected properties. For more information, see [Image property protection](#) in the *OpenStack Cloud Administrator Guide*.

Method	URI	Description
API versions		
GET	/	Lists version information for all Image service API versions.
Images (images)		
GET	/v2/images{?limit,marker,name,visibility,member_status,owner,status,size_min,size_max,sort_key,sort_dir,sort,tag,created_at,updated_at}	(Since Image API v2.0) Lists public virtual machine (VM) images.
POST	/v2/images	(Since Image API v2.0) Creates a virtual machine (VM) image.
GET	/v2/images/{image_id}	(Since Image API v2.0) Shows details for an image.
PATCH	/v2/images/{image_id}	(Since Image API v2.0) Updates an image.
DELETE	/v2/images/{image_id}	(Since Image API v2.0) Deletes an image.
POST	/v2/images/{image_id}/actions/re-activate	(Since Image API v2.0) Reactivates an image.
POST	/v2/images/{image_id}/actions/de-activate	(Since Image API v2.0) Deactivates an image.
Image data (images, file)		
PUT	/v2/images/{image_id}/file	(Since Image API v2.0) Uploads binary image data.
GET	/v2/images/{image_id}/file	(Since Image API v2.0) Downloads binary image data.
Image tags (images, tags)		
PUT	/v2/images/{image_id}/tags/{tag}	(Since Image API v2.0) Adds a tag to an image.
DELETE	/v2/images/{image_id}/tags/{tag}	(Since Image API v2.0) Deletes a tag from an image.
Members (images, members)		
GET	/v2/images/{image_id}/members	(Since Image API v2.1) Lists the tenants that share this image.
POST	/v2/images/{image_id}/members	(Since Image API v2.1) Adds a tenant ID as an image member.
GET	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.2) Shows image member details.
PUT	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.1) Sets the status for an image member.
DELETE	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.1) Deletes a tenant ID from the member list of an image.
Image schemas (schemas, image, images, member, members)		
GET	/v2/schemas/images	(Since Images v2.0) Shows a JSON schema document that represents an images entity.
GET	/v2/schemas/image	(Since Images v2.0) Shows a JSON schema document that represents an image entity.
GET	/v2/schemas/members	(Since Images v2.1) Shows a JSON schema document that represents an image members entity.

Method	URI	Description
GET	/v2/schemas/member	(Since Images v2.1) Shows a JSON schema document that represents an image member entity.
Metadata definition resource types (since API v2.0) (metadefs, namespaces, resource_types)		
GET	/v2/metadefs/resource_types	Lists all resource types.
POST	/v2/metadefs/namespaces/{namespace_id}/resource_types	Creates a resource type association in a namespace.
GET	/v2/metadefs/namespaces/{namespace_id}/resource_types	Lists resource type associations in a namespace.
DELETE	/v2/metadefs/namespaces/{namespace_id}/resource_types/{name}	Removes a resource type association in a namespace.
Metadata definition namespaces (since API v2.0) (metadefs, namespaces)		
POST	/v2/metadefs/namespaces	Creates a namespace.
GET	/v2/metadefs/namespaces{?limit, marker, visibility, resource_types, sort_key, sort_dir}	Lists public namespaces.
GET	/v2/metadefs/namespaces/{namespace_id}	Gets details for a namespace.
PUT	/v2/metadefs/namespaces/{namespace_id}	Updates a namespace.
DELETE	/v2/metadefs/namespaces/{namespace_id}	Deletes a namespace and its properties, objects, and any resource type associations.
Metadata definition properties (since API v2.0) (metadefs, namespaces, properties)		
POST	/v2/metadefs/namespaces/{namespace_id}/properties	Creates a property definition in a namespace.
GET	/v2/metadefs/namespaces/{namespace_id}/properties	Lists property definitions in a namespace.
GET	/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}{?resource_type}	Shows the definition for a property.
PUT	/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}	Updates a property definition.
DELETE	/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}	Removes a property definition in a namespace.
Metadata definition objects (since API v2.0) (metadefs, namespaces, objects)		
POST	/v2/metadefs/namespaces/{namespace_id}/objects	Creates an object definition in a namespace.
GET	/v2/metadefs/namespaces/{namespace_id}/objects{?visibility, resource_types, sort_key, sort_dir}	Lists object definitions in a namespace.
GET	/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}	Shows the definition for an object.
PUT	/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}	Updates an object definition in a namespace.

Method	URI	Description
DELETE	/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}	Deletes an object definition from a namespace.
Metadata definition tags (since API v2.0) (metadefs, namespaces, tags)		
POST	/v2/metadefs/namespaces/tags/{namespace_id}	Creates one or more tag definitions in a namespace.
GET	/v2/metadefs/namespaces/tags/{namespace_id}{?limit,marker,sort_key,sort_dir}	Lists the tag definitions within a namespace.
DELETE	/v2/metadefs/namespaces/tags/{namespace_id}	Deletes all tag definitions within a namespace.
POST	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Adds a tag to the list of namespace tag definitions.
GET	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Gets a definition for a tag.
PUT	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Renames a tag definition.
DELETE	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Deletes a tag definition within a namespace.
Metadata definition schemas (schemas, metadefs, namespace, namespaces, object, objects, property, properties, tag, tags, resource_type, resource_types)		
GET	/v2/schemas/metadefs/namespace	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace entity.
GET	/v2/schemas/metadefs/namespaces	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespaces entity.
GET	/v2/schemas/metadefs/object	(Since API v2.1) Shows a JSON schema document that represents a metadata definition object entity.
GET	/v2/schemas/metadefs/objects	(Since API v2.1) Shows a JSON schema document that represents a metadata definition objects entity.
GET	/v2/schemas/metadefs/property	(Since API v2.1) Shows a JSON schema document that represents a metadata definition property entity.
GET	/v2/schemas/metadefs/properties	(Since API v2.1) Shows a JSON schema document that represents a metadata definition properties entity.
GET	/v2/schemas/metadefs/tag	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tag entity.
GET	/v2/schemas/metadefs/tags	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tags entity.
GET	/v2/schemas/metadefs/resource_type	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type association entity.
GET	/v2/schemas/metadefs/resource_types	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type associations entity.
Tasks (since API v2.2) (tasks)		
POST	/v2/tasks	Creates a task.
GET	/v2/tasks{?sort_dir,sort_key,type,status}	Lists tasks.
GET	/v2/tasks/{task_id}	Shows details for a task.

12.1. API versions

Method	URI	Description
GET	/	Lists version information for all Image service API versions.

12.1.1. List API versions

Method	URI	Description
GET	/	Lists version information for all Image service API versions.

Normal response codes: 300

12.1.1.1. Request

This operation does not accept a request body.

12.2. Images (images)

Creates, lists, updates, and deletes images.

Image operations show all fields in the response body. Any field with no value is set to null value (JSON null data type).

The possible status values for images are:

Table 12.1. Image status

Status	Description
queued	The Image service reserved an image ID for the image in the registry but did not yet upload any image data.
saving	The Image service is currently uploading the raw data for the image.
active	The image is active and fully available in the Image service.
killed	An image data upload error occurred.
deleted	The Image service retains information about the image but the image is no longer available for use.
pending_delete	Similar to the deleted status. An image in this state is not recoverable.

Method	URI	Description
GET	/v2/images{?limit,marker,name,visibility,member_status,owner,status,size_min,size_max,sort_key,sort_dir,sort,tag,created_at,updated_at}	(Since Image API v2.0) Lists public virtual machine (VM) images.
POST	/v2/images	(Since Image API v2.0) Creates a virtual machine (VM) image.
GET	/v2/images/{image_id}	(Since Image API v2.0) Shows details for an image.
PATCH	/v2/images/{image_id}	(Since Image API v2.0) Updates an image.
DELETE	/v2/images/{image_id}	(Since Image API v2.0) Deletes an image.
POST	/v2/images/{image_id}/actions/reactivate	(Since Image API v2.0) Reactivates an image.
POST	/v2/images/{image_id}/actions/deactivate	(Since Image API v2.0) Deactivates an image.

12.2.1. List images

Method	URI	Description
GET	/v2/images{?limit,marker,name,visibility,member_status,owner,status,size_min,size_max,sort_key,sort_dir,sort,tag,created_at,updated_at}	(Since Image API v2.0) Lists public virtual machine (VM) images.

Returns a subset of the larger collection of images and a link that you can use to get the next set of images. You should always check for the presence of a next link and use it as the URI in a subsequent HTTP GET request. You should follow this pattern until a next link is no longer provided. The next link preserves any query parameters that you send in your initial request. You can use the `first` link to jump back to the first page of the collection. If you prefer to paginate through images manually, use the `limit` and `marker` parameters.

The list operation accepts query parameters to filter the response.

A client can provide direct comparison filters by using most image attributes, such as `name=Ubuntu`, `visibility=public`, and so on. A client cannot use tags or any link in the json-schema, such as `self`, `file`, or `schema`, to filter the response.

You can use the `size_min` and `size_max` query parameters to filter images that are greater than or less than the image size. The size, in bytes, is the size of an image on disk.

For example, to filter the container to include only images that are from 1 to 4 MB, set the `size_min` query parameter to 1048576 and the `size_max` query parameter to 4194304.

You can list VM images that have a status of `active`, `queued`, or `saving`.

You can use query parameters to sort the results of this operation.

- `sort_key`. Sorts by an image attribute. Sorts in the natural sorting direction of the image attribute.
- `sort_dir`. Sorts in a sort direction.
- `sort`. Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, the default is `desc`.

To sort the response, use the `sort_key` and `sort_dir` query parameters:

```
GET /v2/images?sort_key=name&sort_dir=asc&sort_key=status&sort_dir=desc
```

Alternatively, specify the `sort` query parameter:

```
GET /v2/images?sort=name:asc,status:desc
```

Normal response codes: 200

12.2.1.1. Request

This table shows the query parameters for the list images request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
name	String (Optional)	Filters the response by a name, as a string. A valid value is the name of an image.
visibility	String (Optional)	Filters the response by an image visibility value or values. Use the comma (,) character to separate multiple values. A valid value is <code>public</code> , <code>private</code> , or <code>shared</code> . If you omit this parameter, the response shows <code>public</code> , <code>private</code> , and <code>shared</code> images with a member status of <code>accepted</code> .
member_status	String (Optional)	Filters the response by a member status. A valid value is <code>accepted</code> , <code>pending</code> , <code>rejected</code> , or <code>all</code> . Default is <code>accepted</code> .
owner	String (Optional)	Filters the response by a tenant ID. Shows only images that are shared with this tenant owner.
status	Int (Optional)	Filters the response by an image status. A valid value is <code>queued</code> , <code>saving</code> , <code>active</code> , <code>killed</code> , <code>deleted</code> , or <code>pending_delete</code> .
size_min	String (Optional)	Filters the response by a minimum image size, in bytes.
size_max	String (Optional)	Filters the response by a maximum image size, in bytes.
sort_key	String (Optional)	Sorts the response by an attribute, such as <code>name</code> , <code>id</code> , or <code>updated_at</code> . Default is <code>created_at</code> . The API uses the natural sorting direction of the <code>sort_key</code> image attribute.
sort_dir	String (Optional)	Sorts the response by a set of one or more sort direction and attribute (<code>sort_key</code>) combinations. A valid value for the sort direction is <code>asc</code> (ascending) or <code>desc</code> (descending). If you omit the sort direction in a set, the default is <code>desc</code> .
sort	String (Optional)	Sorts the response by one or more attribute and sort direction combinations. You can also set multiple sort keys and directions. Default direction is <code>desc</code> . Use the comma (,) character to separate multiple values. For example: <code>GET /v2/images?sort=name:asc,status:desc</code>
tag	String (Optional)	Filters the response by an image tag. For example, <code>?tag="cirros"</code> .
created_at	DateTime (Optional)	Filters the response by the date and time when the resource was created. The date and time stamp format is ISO 8601 : <code>CCYY-MM-DDThh:mm:ss±hh:mm</code>

Name	Type	Description
		The $\pm hh:mm$ value, if included, returns the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. If you omit the <code>created_at</code> date and time stamp, its value is null.
<code>updated_at</code>	DateTime (Optional)	Filters the response by the date and time when the resource was updated. The date and time stamp format is ISO 8601: <code>CCYY-MM-DDThh:mm:ss±hh:mm</code> The $\pm hh:mm$ value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. If you omit the time zone, the UTC time zone is assumed. If you omit the <code>updated_at</code> date and time stamp, its value is null.

This operation does not accept a request body.

12.2.1.2. Response

Example 12.1. List images: JSON response

```
{
  "images": [
    {
      "status": "active",
      "name": "cirros-0.3.2-x86_64-disk",
      "tags": [],
      "container_format": "bare",
      "created_at": "2014-11-07T17:07:06Z",
      "disk_format": "qcow2",
      "updated_at": "2014-11-07T17:19:09Z",
      "visibility": "public",
      "self": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27",
      "min_disk": 0,
      "protected": false,
      "id": "1bea47ed-f6a9-463b-b423-14b9cca9ad27",
      "file": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27/file",
      "checksum": "64d7c1cd2b6f60c92c14662941cb7913",
      "owner": "5ef70662f8b34079a6eddb8da9d75fe8",
      "size": 13167616,
      "min_ram": 0,
      "schema": "/v2/schemas/image",
      "virtual_size": null
    },
    {
      "status": "active",
      "name": "F17-x86_64-cfntools",
      "tags": [],
      "container_format": "bare",
      "created_at": "2014-10-30T08:23:39Z",
      "disk_format": "qcow2",
      "updated_at": "2014-11-03T16:40:10Z",
      "visibility": "public",
      "self": "/v2/images/781b3762-9469-4cec-b58d-3349e5de4e9c",
      "min_disk": 0,

```

```
        "protected": false,  
        "id": "781b3762-9469-4cec-b58d-3349e5de4e9c",  
        "file": "/v2/images/781b3762-9469-4cec-b58d-3349e5de4e9c/file",  
        "checksum": "afab0f79bac770d61d24b4d0560b5f70",  
        "owner": "5ef70662f8b34079a6eddb8da9d75fe8",  
        "size": 476704768,  
        "min_ram": 0,  
        "schema": "/v2/schemas/image",  
        "virtual_size": null  
    },  
    ],  
    "schema": "/v2/schemas/images",  
    "first": "/v2/images"  
}
```

12.2.2. Create image

Method	URI	Description
POST	/v2/images	(Since Image API v2.0) Creates a virtual machine (VM) image.

The Location response header contains the URI for the image. The response body contains the new image entity.

Synchronous Postconditions

- With correct permissions, you can see the image status as queued through API calls.

Normal response codes: 201

12.2.2.1. Request

Example 12.2. Create image: JSON request

```
{
  "container_format": "bare",
  "disk_format": "raw",
  "name": "Ubuntu",
  "id": "b2173dd3-7ad6-4362-baa6-a68bce3565cb"
}
```

12.2.2.2. Response

Example 12.3. Create image: JSON response

```
{
  "status": "queued",
  "name": "Ubuntu",
  "tags": [],
  "container_format": "bare",
  "created_at": "2015-11-29T22:21:42Z",
  "size": null,
  "disk_format": "raw",
  "updated_at": "2015-11-29T22:21:42Z",
  "visibility": "private",
  "locations": [],
  "self": "/v2/images/b2173dd3-7ad6-4362-baa6-a68bce3565cb",
  "min_disk": 0,
  "protected": false,
  "id": "b2173dd3-7ad6-4362-baa6-a68bce3565cb",
  "file": "/v2/images/b2173dd3-7ad6-4362-baa6-a68bce3565cb/file",
  "checksum": null,
  "owner": "bab7d5c60cd041a0a36f7c4b6e1dd978",
  "virtual_size": null,
  "min_ram": 0,
  "schema": "/v2/schemas/image"
}
```

12.2.3. Show image details

Method	URI	Description
GET	/v2/images/{image_id}	(Since Image API v2.0) Shows details for an image.

The response body contains a single image entity.

Preconditions

- The image must exist.

Normal response codes: 200

Error response codes: itemNotFound (404)

12.2.3.1. Request

This table shows the URI parameters for the show image details request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

12.2.3.2. Response

Example 12.4. Show image details: JSON response

```
{
  "status": "active",
  "name": "cirros-0.3.2-x86_64-disk",
  "tags": [],
  "container_format": "bare",
  "created_at": "2014-05-05T17:15:10Z",
  "disk_format": "qcow2",
  "updated_at": "2014-05-05T17:15:11Z",
  "visibility": "public",
  "self": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27",
  "min_disk": 0,
  "protected": false,
  "id": "1bea47ed-f6a9-463b-b423-14b9cca9ad27",
  "file": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27/file",
  "checksum": "64d7c1cd2b6f60c92c14662941cb7913",
  "owner": "5ef70662f8b34079a6eddb8da9d75fe8",
  "size": 13167616,
  "min_ram": 0,
  "schema": "/v2/schemas/image",
  "virtual_size": null
}
```

12.2.4. Update image

Method	URI	Description
PATCH	/v2/images/{image_id}	(Since Image API v2.0) Updates an image.

Depending on the referenced target location, this operation performs one of these actions:

Table 12.2. Image update actions

Target location	Update action
An array index	The API inserts a new value into the array at the index.
An object member that does not exist	The API adds a member to the object.
An object member that exists	The member value is replaced.

The operation object must contain a `value` member that contains the value to add. For example:

```
{
  "op": "add",
  "path": "/a/b/c",
  "value": [
    "foo",
    "bar"
  ]
}
```

The target location must reference one of these values:

- The root of the target document. The value is the entire content of the target document.
- A member value to add to an object. The API adds the value to the object at the location. If the member already exists, the API replaces it with the value.
- An element to add to the array. The API adds the element value to the array at the location. The API shifts any element that is at or above the index one position to the right. The index must not be greater than the number of elements in the array. If you use the hyphen (-) character to index the end of the array, the API appends the value to the array. See [JavaScript Object Notation \(JSON\) Pointer](#).

Because this operation adds to existing objects and arrays, its target location often does not exist.

The request body must conform to one of these media types:

- `application/openstack-images-v2.0-json-patch`
- `application/openstack-images-v2.1-json-patch` (since Image API v2.2)

You can also use the **PATCH** method to add or remove image properties.

For information about the **PATCH** method and the available media types, see [Image API v2 HTTP PATCH media types](#).

Preconditions

- When you add a location to or replace a location in the image, you must set the `disk_format` and `container_format` parameters in the image.
- When you replace a location, that location must be previously set in the image.

Synchronous Postconditions

- With correct permissions, you can view the updated values of the attributes of the image.
- After you add a location to an image that had no location and with correct permissions, you can use API calls to view the image status as `active`.
- After you remove all locations from the image and with correct permissions, you can use API calls to view the image status as `queued`.

Troubleshooting

- If you cannot update locations, your request might be missing some information. Make sure that you meet the preconditions and run the request again. If the request fails again, review your API request.

Normal response codes: 200

Error response codes: `itemNotFound` (404)

12.2.4.1. Request

This table shows the URI parameters for the update image request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

Example 12.5. Update image: JSON request

```
[
  {
    "op": "replace",
    "path": "/name",
    "value": "Fedora 17"
  },
  {
    "op": "replace",
    "path": "/tags",
    "value": [
      "fedora",
      "beefy"
    ]
  }
]
```

12.2.4.2. Response

Example 12.6. Update image: JSON response

```
{
```



```
"id": "da3b75d9-3f4a-40e7-8a2c-bfab23927dea",
"name": "Fedora 17",
"status": "active",
"visibility": "public",
"size": 2254249,
"checksum": "2cec138d7dae2aa59038ef8c9aec2390",
"tags": [
  "fedora",
  "beefy"
],
"created_at": "2012-08-10T19:23:50Z",
"updated_at": "2012-08-12T11:11:33Z",
"self": "/v2/images/da3b75d9-3f4a-40e7-8a2c-bfab23927dea",
"file": "/v2/images/da3b75d9-3f4a-40e7-8a2c-bfab23927dea/file",
"schema": "/v2/schemas/image",
"owner": null,
"min_ram": null,
"min_disk": null,
"disk_format": null,
"virtual_size": null,
"container_format": null
}
```

12.2.5. Delete image

Method	URI	Description
DELETE	/v2/images/{image_id}	(Since Image API v2.0) Deletes an image.

You cannot delete images with the protected attribute set to `true` (boolean).

Preconditions

- You can delete an image in any status except `deleted`.
- First, set the protected attribute to `false` (boolean). Then, perform the delete.

Synchronous Postconditions

- The response is empty and returns the HTTP 204 response code.
- The API deletes the image from the images index.
- If the image stores binary image data in the storage node, the OpenStack Image service deletes the data from the node.

Troubleshooting

- The call returns the HTTP 403 response code when the protected attribute is set to `true` even if you have correct permissions. Ensure that you meet the preconditions and run the request again. If the request fails again, review your API request.

Normal response codes: 204

Error response codes: forbidden (403), itemNotFound (404)

12.2.5.1. Request

This table shows the URI parameters for the delete image request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

12.2.6. Reactivate image

Method	URI	Description
POST	/v2/images/{image_id}/actions/re-activate	(Since Image API v2.0) Reactivates an image.

The reactivate operation returns an error if the image status is not active or deactivated.

Preconditions

- The image must exist.

Normal response codes: 204

Error response codes: itemNotFound (404)

12.2.6.1. Request

This table shows the URI parameters for the reactivate image request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

12.2.7. Deactivate image

Method	URI	Description
POST	/v2/images/{image_id}/actions/de-activate	(Since Image API v2.0) Deactivates an image.

If you try to download a deactivated image, the call returns the Forbidden (403) response code. Also, only administrative users can view image locations for deactivated images.

The deactivate operation returns an error if the image status is not active or deactivated.

Preconditions

- The image must exist.

Normal response codes: 204

Error response codes: forbidden (403), itemNotFound (404)

12.2.7.1. Request

This table shows the URI parameters for the deactivate image request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

12.3. Image data (images, file)

Uploads and downloads raw image data.

Method	URI	Description
PUT	/v2/images/{image_id}/file	(Since Image API v2.0) Uploads binary image data.
GET	/v2/images/{image_id}/file	(Since Image API v2.0) Downloads binary image data.

12.3.1. Upload binary image data

Method	URI	Description
PUT	/v2/images/{image_id}/file	(Since Image API v2.0) Uploads binary image data.

Set the Content-Type request header to `application/octet-stream`.

Example call:

```
curl -i -X PUT -H "X-Auth-Token: $token" -H "Content-Type: application/octet-stream" \
  -d @/home/glance/ubuntu-12.10.qcow2 $image_url/v2/images/{image_id}/file
```

Preconditions

Before you can store binary image data, you must meet the following preconditions:

- The image must exist.
- You must set the disk and container formats in the image.
- The image status must be queued.
- Your image storage quota must be sufficient.
- The size of the data that you want to store must not exceed the size that the OpenStack Image service allows.

Synchronous Postconditions

- With correct permissions, you can see the image status as `active` through API calls.
- With correct access, you can see the stored data in the storage system that OpenStack Image service manages.

Troubleshooting

- If you cannot store the data, either your request lacks required information or you exceeded your allotted quota. Ensure that you meet the preconditions and run the request again. If the request fails again, review your API request.
- The storage back ends for storing the data must have enough free storage space to accommodate the size of the data.

Normal response codes: 204

12.3.1.1. Request

This table shows the URI parameters for the upload binary image data request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

12.3.2. Download binary image data

Method	URI	Description
GET	/v2/images/{image_id}/file	(Since Image API v2.0) Downloads binary image data.

Example call: `curl -i -X GET -H "X-Auth-Token: $token" $image_url/v2/images/{image_id}/file`

The response body contains the raw binary data that represents the actual virtual disk. The Content-Type header contains the application/octet-stream value. The Content-MD5 header contains an MD5 checksum of the image data. Use this checksum to verify the integrity of the image data.

Preconditions

- The images must exist.

Synchronous Postconditions

- You can download the binary image data in your machine if the image has image data.
- If image data exists, the call returns the HTTP 200 response code.
- If no image data exists, the call returns the HTTP 204 response code.

Normal response codes: 200204

Error response codes: forbidden (403), itemNotFound (404)

12.3.2.1. Request

This table shows the header parameters for the download binary image data request:

Name	Type	Description
Content-Range	String (Optional)	The content range of image data. For details, see Hypertext Transfer Protocol (HTTP/1.1): Range Requests .

This table shows the URI parameters for the download binary image data request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

12.4. Image tags (images, tags)

Adds and deletes image tags.

Method	URI	Description
PUT	/v2/images/{image_id}/tags/{tag}	(Since Image API v2.0) Adds a tag to an image.

Method	URI	Description
DELETE	/v2/images/{image_id}/tags/{tag}	(Since Image API v2.0) Deletes a tag from an image.

12.4.1. Add image tag

Method	URI	Description
PUT	/v2/images/{image_id}/tags/{tag}	(Since Image API v2.0) Adds a tag to an image.

Normal response codes: 204

Error response codes: itemNotFound (404)

12.4.1.1. Request

This table shows the URI parameters for the add image tag request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.
{tag}	String	The image tag.

This operation does not accept a request body.

12.4.2. Delete image tag

Method	URI	Description
DELETE	/v2/images/{image_id}/tags/{tag}	(Since Image API v2.0) Deletes a tag from an image.

Normal response codes: 204

Error response codes: itemNotFound (404)

12.4.2.1. Request

This table shows the URI parameters for the delete image tag request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.
{tag}	String	The image tag.

This operation does not accept a request body.

12.5. Members (images, members)

Creates, lists, updates, and deletes image members.

Method	URI	Description
GET	/v2/images/{image_id}/members	(Since Image API v2.1) Lists the tenants that share this image.
POST	/v2/images/{image_id}/members	(Since Image API v2.1) Adds a tenant ID as an image member.
GET	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.2) Shows image member details.
PUT	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.1) Sets the status for an image member.
DELETE	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.1) Deletes a tenant ID from the member list of an image.

12.5.1. List image members

Method	URI	Description
GET	/v2/images/{image_id}/members	(Since Image API v2.1) Lists the tenants that share this image.

If a user who shares this image makes this call, the member list contains only information for that user.

If a user who does not share this image makes this call, the call returns the HTTP 404 response code.

Preconditions

- The image must exist.
- You must be the owner or a member of the image.

Normal response codes: 200

Error response codes: itemNotFound (404)

12.5.1.1. Request

This table shows the URI parameters for the list image members request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

This operation does not accept a request body.

12.5.1.2. Response

Example 12.7. List image members: JSON response

```
{
  "members": [
    {
      "created_at": "2013-10-07T17:58:03Z",
      "image_id": "dbc999e3-c52f-4200-bedd-3b18fe7f87fe",
      "member_id": "123456789",
      "schema": "/v2/schemas/member",
      "status": "pending",
      "updated_at": "2013-10-07T17:58:03Z"
    },
    {
      "created_at": "2013-10-07T17:58:55Z",
      "image_id": "dbc999e3-c52f-4200-bedd-3b18fe7f87fe",
      "member_id": "987654321",
      "schema": "/v2/schemas/member",
      "status": "accepted",
      "updated_at": "2013-10-08T12:08:55Z"
    }
  ],
}
```

```
"schema": "/v2/schemas/members"  
}
```

12.5.2. Create image member

Method	URI	Description
POST	/v2/images/{image_id}/members	(Since Image API v2.1) Adds a tenant ID as an image member.

This call accepts either the `member_id` or `member` attribute in the request body. If you specify both attributes, the API uses the `member_id` value and ignores the `member` value. Use of the `member` attribute is supported but deprecated. Use of the `member_id` attribute is preferred.

Preconditions

- The images must exist.
- You can add a member to a private image.
- You must be the owner of the image.

Synchronous Postconditions

- With correct permissions, you can see the member status of the image as pending through API calls.

Troubleshooting

- Even if you have correct permissions, if the `visibility` attribute is set to `public`, the request returns the HTTP 403 response code. Ensure that you meet the preconditions and run the request again. If the request fails again, review your API request.
- If the member is already a member for the image, the service returns the `Conflict` (409) response code. If you meant to specify a different member, run the request again.

Normal response codes: 200

Error response codes: `invalidVisibility` (403), `memberConflict` (409)

12.5.2.1. Request

This table shows the URI parameters for the create image member request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.

Example 12.8. Create image member: JSON request

```
{
  "member": "8989447062e04a818baf9e073fd04fa7"
}
```

12.5.2.2. Response

Example 12.9. Create image member: JSON response

```
{
  "created_at": "2013-09-20T19:22:19Z",
  "image_id": "a96be11e-8536-4910-92cb-de50aa19dfe6",
  "member_id": "8989447062e04a818baf9e073fd04fa7",
  "schema": "/v2/schemas/member",
  "status": "pending",
  "updated_at": "2013-09-20T19:25:31Z"
}
```

12.5.3. Show image member details

Method	URI	Description
GET	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.2) Shows image member details.

Response body is a single image member entity.

Preconditions

- The image must exist.
- You must be the owner or a member of the image.

Normal response codes: 200

12.5.3.1. Request

This table shows the URI parameters for the show image member details request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.
{member_id}	String	The ID of the image member. An image member is a tenant with whom the image is shared.

This operation does not accept a request body.

12.5.3.2. Response

Example 12.10. Show image member details: JSON response

```
{
  "status": "pending",
  "created_at": "2013-11-26T07:21:21Z",
  "updated_at": "2013-11-26T07:21:21Z",
  "image_id": "0ae74cc5-5147-4239-9ce2-b0c580f7067e",
  "member_id": "8989447062e04a818baf9e073fd04fa7",
  "schema": "/v2/schemas/member"
}
```

12.5.4. Update image member

Method	URI	Description
PUT	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.1) Sets the status for an image member.

Preconditions

- The images must exist.
- You must be a member of the image.

Synchronous Postconditions

- If you update the member status to accepted and have the correct permissions, you see the image in list images responses.
- With correct permissions, you can make API calls to see the updated member status of the image.

Normal response codes: 200

12.5.4.1. Request

This table shows the URI parameters for the update image member request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.
{member_id}	String	The ID of the image member. An image member is a tenant with whom the image is shared.

Example 12.11. Update image member: JSON request

```
{
  "status": "accepted"
}
```

12.5.4.2. Response

Example 12.12. Update image member: JSON response

```
{
  "created_at": "2013-09-20T19:22:19Z",
  "image_id": "a96be11e-8536-4910-92cb-de50aa19dfe6",
  "member_id": "8989447062e04a818baf9e073fd04fa7",
  "schema": "/v2/schemas/member",
  "status": "accepted",
  "updated_at": "2013-09-20T20:15:31Z"
}
```

12.5.5. Delete image member

Method	URI	Description
DELETE	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.1) Deletes a tenant ID from the member list of an image.

Preconditions

- The image must exist.
- You must be the owner of the image.

Synchronous Postconditions

- The API removes the member from the image members.

Troubleshooting

- Even if you have correct permissions, if you are not the owner of the image or you specify an incorrect image ID or member ID, the call returns the HTTP 403 or 404 response code. Ensure that you meet the preconditions and run the request again. If the request fails again, review your API request URI.

Normal response codes: 204

Error response codes: forbidden (403), itemNotFound (404)

12.5.5.1. Request

This table shows the URI parameters for the delete image member request:

Name	Type	Description
{image_id}	UUID	The UUID of the image.
{member_id}	String	The ID of the image member. An image member is a tenant with whom the image is shared.

This operation does not accept a request body.

12.6. Image schemas (schemas, image, images, member, members)

Gets a JSON-schema document that represents an images or image entity.

Method	URI	Description
GET	/v2/schemas/images	(Since Images v2.0) Shows a JSON schema document that represents an images entity.
GET	/v2/schemas/image	(Since Images v2.0) Shows a JSON schema document that represents an image entity.
GET	/v2/schemas/members	(Since Images v2.1) Shows a JSON schema document that represents an image members entity.

Method	URI	Description
GET	/v2/schemas/member	(Since Images v2.1) Shows a JSON schema document that represents an image member entity.

12.6.1. Show images schema

Method	URI	Description
GET	/v2/schemas/images	(Since Images v2.0) Shows a JSON schema document that represents an images entity.

An images entity is a container of image entities.

The following schema is solely an example. Consider only the response to the API call as authoritative.

Normal response codes: 200

12.6.1.1. Request

This operation does not accept a request body.

12.6.1.2. Response

Example 12.13. Show images schema: JSON response

```
{
  "links": [
    {
      "href": "{first}",
      "rel": "first"
    },
    {
      "href": "{next}",
      "rel": "next"
    },
    {
      "href": "{schema}",
      "rel": "describedby"
    }
  ],
  "name": "images",
  "properties": {
    "first": {
      "type": "string"
    },
    "images": {
      "items": {
        "additionalProperties": {
          "type": "string"
        },
        "links": [
          {
            "href": "{self}",
            "rel": "self"
          },
          {
            "href": "{file}",
            "rel": "enclosure"
          },
          {
            "href": "{schema}",

```

```

        "rel": "describedby"
    },
    ],
    "name": "image",
    "properties": {
        "architecture": {
            "description": "Operating system architecture as
specified in http://docs.openstack.org/trunk/openstack-compute/admin/content/
adding-images.html",
            "is_base": false,
            "type": "string"
        },
        "checksum": {
            "description": "md5 hash of image contents. (READ-
ONLY)",
            "maxLength": 32,
            "type": [
                "null",
                "string"
            ]
        },
        "container_format": {
            "description": "Format of the container",
            "enum": [
                null,
                "ami",
                "ari",
                "aki",
                "bare",
                "ovf",
                "ova"
            ],
            "type": [
                "null",
                "string"
            ]
        },
        "created_at": {
            "description": "Date and time of image registration
(READ-ONLY)",
            "type": "string"
        },
        "direct_url": {
            "description": "URL to access the image file kept in
external store (READ-ONLY)",
            "type": "string"
        },
        "disk_format": {
            "description": "Format of the disk",
            "enum": [
                null,
                "ami",
                "ari",
                "aki",
                "vhd",
                "vmdk",
                "raw",
                "qcow2",
                "vdi",
                "iso"
            ]
        }
    }
}

```

```

        ],
        "type": [
            "null",
            "string"
        ]
    },
    "file": {
        "description": "(READ-ONLY)",
        "type": "string"
    },
    "id": {
        "description": "An identifier for the image",
        "pattern": "^[0-9a-fA-F]{8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
        "type": "string"
    },
    "instance_uuid": {
        "description": "ID of instance used to create this
image.",
        "is_base": false,
        "type": "string"
    },
    "kernel_id": {
        "description": "ID of image stored in Glance that
should be used as the kernel when booting an AMI-style image.",
        "is_base": false,
        "pattern": "^[0-9a-fA-F]{8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
        "type": "string"
    },
    "locations": {
        "description": "A set of URLs to access the image file
kept in external store",
        "items": {
            "properties": {
                "metadata": {
                    "type": "object"
                },
                "url": {
                    "maxLength": 255,
                    "type": "string"
                }
            },
            "required": [
                "url",
                "metadata"
            ],
            "type": "object"
        },
        "type": "array"
    },
    "min_disk": {
        "description": "Amount of disk space (in GB) required
to boot image.",
        "type": "integer"
    },
    "min_ram": {
        "description": "Amount of ram (in MB) required to boot
image.",
        "type": "integer"
    }

```

```

    },
    "name": {
      "description": "Descriptive name for the image",
      "maxLength": 255,
      "type": [
        "null",
        "string"
      ]
    },
    "os_distro": {
      "description": "Common name of operating system
distribution as specified in http://docs.openstack.org/trunk/openstack-
compute/admin/content/adding-images.html",
      "is_base": false,
      "type": "string"
    },
    "os_version": {
      "description": "Operating system version as specified
by the distributor",
      "is_base": false,
      "type": "string"
    },
    "owner": {
      "description": "Owner of the image",
      "maxLength": 255,
      "type": [
        "null",
        "string"
      ]
    },
    "protected": {
      "description": "If true, image will not be deletable.",
      "type": "boolean"
    },
    "ramdisk_id": {
      "description": "ID of image stored in Glance that
should be used as the ramdisk when booting an AMI-style image.",
      "is_base": false,
      "pattern": "^[0-9a-fA-F]{8}-([0-9a-fA-F]){4}-([0-9a-
fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
      "type": "string"
    },
    "schema": {
      "description": "(READ-ONLY)",
      "type": "string"
    },
    "self": {
      "description": "(READ-ONLY)",
      "type": "string"
    },
    "size": {
      "description": "Size of image file in bytes (READ-
ONLY)",
      "type": [
        "null",
        "integer"
      ]
    },
    "status": {

```

```

        "description": "Status of the image (READ-ONLY)",
        "enum": [
            "queued",
            "saving",
            "active",
            "killed",
            "deleted",
            "pending_delete"
        ],
        "type": "string"
    },
    "tags": {
        "description": "List of strings related to the image",
        "items": {
            "maxLength": 255,
            "type": "string"
        },
        "type": "array"
    },
    "updated_at": {
        "description": "Date and time of the last image
modification (READ-ONLY)",
        "type": "string"
    },
    "virtual_size": {
        "description": "Virtual size of image in bytes (READ-
ONLY)",
        "type": [
            "null",
            "integer"
        ]
    },
    "visibility": {
        "description": "Scope of image accessibility",
        "enum": [
            "public",
            "private"
        ],
        "type": "string"
    }
}

},
"type": "array"
},
"next": {
    "type": "string"
},
"schema": {
    "type": "string"
}
}
}

```

12.6.2. Show image schema

Method	URI	Description
GET	/v2/schemas/image	(Since Images v2.0) Shows a JSON schema document that represents an image entity.

The following schema is solely an example. Consider only the response to the API call as authoritative.

Normal response codes: 200

12.6.2.1. Request

This operation does not accept a request body.

12.6.2.2. Response

Example 12.14. Show image schema: JSON response

```
{
  "additionalProperties": {
    "type": "string"
  },
  "links": [
    {
      "href": "{self}",
      "rel": "self"
    },
    {
      "href": "{file}",
      "rel": "enclosure"
    },
    {
      "href": "{schema}",
      "rel": "describedby"
    }
  ],
  "name": "image",
  "properties": {
    "architecture": {
      "description": "Operating system architecture as specified in
http://docs.openstack.org/trunk/openstack-compute/admin/content/adding-
images.html",
      "is_base": false,
      "type": "string"
    },
    "checksum": {
      "description": "md5 hash of image contents. (READ-ONLY)",
      "maxLength": 32,
      "type": [
        "null",
        "string"
      ]
    },
    "container_format": {
      "description": "Format of the container",
      "enum": [
```

```

        null,
        "ami",
        "ari",
        "aki",
        "bare",
        "ovf",
        "ova"
    ],
    "type": [
        "null",
        "string"
    ]
},
"created_at": {
    "description": "Date and time of image registration (READ-ONLY)",
    "type": "string"
},
"direct_url": {
    "description": "URL to access the image file kept in external
store (READ-ONLY)",
    "type": "string"
},
"disk_format": {
    "description": "Format of the disk",
    "enum": [
        null,
        "ami",
        "ari",
        "aki",
        "vhd",
        "vmdk",
        "raw",
        "qcow2",
        "vdi",
        "iso"
    ],
    "type": [
        "null",
        "string"
    ]
},
"file": {
    "description": "(READ-ONLY)",
    "type": "string"
},
"id": {
    "description": "An identifier for the image",
    "pattern": "^[0-9a-fA-F]{8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-
([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
    "type": "string"
},
"instance_uuid": {
    "description": "ID of instance used to create this image.",
    "is_base": false,
    "type": "string"
},
"kernel_id": {
    "description": "ID of image stored in Glance that should be used
as the kernel when booting an AMI-style image.",
    "is_base": false,

```



```

        "pattern": "^[([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12})$",
        "type": "string"
    },
    "locations": {
        "description": "A set of URLs to access the image file kept in external store",
        "items": {
            "properties": {
                "metadata": {
                    "type": "object"
                },
                "url": {
                    "maxLength": 255,
                    "type": "string"
                }
            },
            "required": [
                "url",
                "metadata"
            ],
            "type": "object"
        },
        "type": "array"
    },
    "min_disk": {
        "description": "Amount of disk space (in GB) required to boot image.",
        "type": "integer"
    },
    "min_ram": {
        "description": "Amount of ram (in MB) required to boot image.",
        "type": "integer"
    },
    "name": {
        "description": "Descriptive name for the image",
        "maxLength": 255,
        "type": [
            "null",
            "string"
        ]
    },
    "os_distro": {
        "description": "Common name of operating system distribution as specified in http://docs.openstack.org/trunk/openstack-compute/admin/content/adding-images.html",
        "is_base": false,
        "type": "string"
    },
    "os_version": {
        "description": "Operating system version as specified by the distributor",
        "is_base": false,
        "type": "string"
    },
    "owner": {
        "description": "Owner of the image",
        "maxLength": 255,
        "type": [
            "null",

```

```

        "string"
    ]
},
"protected": {
    "description": "If true, image will not be deletable.",
    "type": "boolean"
},
"ramdisk_id": {
    "description": "ID of image stored in Glance that should be used
as the ramdisk when booting an AMI-style image.",
    "is_base": false,
    "pattern": "^[0-9a-fA-F]{8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-
([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
    "type": "string"
},
"schema": {
    "description": "(READ-ONLY)",
    "type": "string"
},
"self": {
    "description": "(READ-ONLY)",
    "type": "string"
},
"size": {
    "description": "Size of image file in bytes (READ-ONLY)",
    "type": [
        "null",
        "integer"
    ]
},
"status": {
    "description": "Status of the image (READ-ONLY)",
    "enum": [
        "queued",
        "saving",
        "active",
        "killed",
        "deleted",
        "pending_delete"
    ],
    "type": "string"
},
"tags": {
    "description": "List of strings related to the image",
    "items": {
        "maxLength": 255,
        "type": "string"
    },
    "type": "array"
},
"updated_at": {
    "description": "Date and time of the last image modification
(READ-ONLY)",
    "type": "string"
},
"virtual_size": {
    "description": "Virtual size of image in bytes (READ-ONLY)",
    "type": [
        "null",
        "integer"
    ]
}

```

```
    ]
  },
  "visibility": {
    "description": "Scope of image accessibility",
    "enum": [
      "public",
      "private"
    ],
    "type": "string"
  }
}
```

12.6.3. Show image members schema

Method	URI	Description
GET	/v2/schemas/members	(Since Images v2.1) Shows a JSON schema document that represents an image members entity.

An image members entity is a container of image member entities.

The following schema is solely an example. Consider only the response to the API call as authoritative.

Normal response codes: 200

12.6.3.1. Request

This operation does not accept a request body.

12.6.3.2. Response

Example 12.15. Show image members schema: JSON response

```
{
  "links": [
    {
      "href": "{schema}",
      "rel": "describedby"
    }
  ],
  "name": "members",
  "properties": {
    "members": {
      "items": {
        "name": "member",
        "properties": {
          "created_at": {
            "description": "Date and time of image member
creation",
            "type": "string"
          },
          "image_id": {
            "description": "An identifier for the image",
            "pattern": "^[0-9a-fA-F]{8}-([0-9a-fA-F]){4}-([0-9a-
fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
            "type": "string"
          },
          "member_id": {
            "description": "An identifier for the image member
(tenantId)",
            "type": "string"
          },
          "schema": {
            "type": "string"
          },
          "status": {
            "description": "The status of this image member",
            "enum": [
```

```
        "pending",
        "accepted",
        "rejected"
    ],
    "type": "string"
},
"updated_at": {
    "description": "Date and time of last modification of
image member",
    "type": "string"
}
}
},
"type": "array"
},
"schema": {
    "type": "string"
}
}
}
```

12.6.4. Show image member schema

Method	URI	Description
GET	/v2/schemas/member	(Since Images v2.1) Shows a JSON schema document that represents an image member entity.

The following schema is solely an example. Consider only the response to the API call as authoritative.

Normal response codes: 200

12.6.4.1. Request

This operation does not accept a request body.

12.6.4.2. Response

Example 12.16. Show image member schema: JSON response

```
{
  "name": "member",
  "properties": {
    "created_at": {
      "description": "Date and time of image member creation",
      "type": "string"
    },
    "image_id": {
      "description": "An identifier for the image",
      "pattern": "^[0-9a-fA-F]{8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
      "type": "string"
    },
    "member_id": {
      "description": "An identifier for the image member (tenantId)",
      "type": "string"
    },
    "schema": {
      "type": "string"
    },
    "status": {
      "description": "The status of this image member",
      "enum": [
        "pending",
        "accepted",
        "rejected"
      ],
      "type": "string"
    },
    "updated_at": {
      "description": "Date and time of last modification of image member",
      "type": "string"
    }
  }
}
```

12.7. Metadata definition resource types (since API v2.0) (metadefs, namespaces, resource_types)

Lists resource types. Also, creates, lists, and removes resource type associations in a namespace.

Method	URI	Description
GET	/v2/metadefs/resource_types	Lists all resource types.
POST	/v2/metadefs/namespaces/{namespace_id}/resource_types	Creates a resource type association in a namespace.
GET	/v2/metadefs/namespaces/{namespace_id}/resource_types	Lists resource type associations in a namespace.
DELETE	/v2/metadefs/namespaces/{namespace_id}/resource_types/{name}	Removes a resource type association in a namespace.

12.7.1. List resource types

Method	URI	Description
GET	/v2/metadefs/resource_types	Lists all resource types.

You can assign metadata definition namespaces to these resource types. See the metadata definition resource types section.

Normal response codes: 200

12.7.1.1. Request

This operation does not accept a request body.

12.7.1.2. Response

Example 12.17. List resource types: JSON response

```
{
  "resource_types": [
    {
      "created_at": "2014-08-28T18:13:04Z",
      "name": "OS::Glance::Image",
      "updated_at": "2014-08-28T18:13:04Z"
    },
    {
      "created_at": "2014-08-28T18:13:04Z",
      "name": "OS::Cinder::Volume",
      "updated_at": "2014-08-28T18:13:04Z"
    },
    {
      "created_at": "2014-08-28T18:13:04Z",
      "name": "OS::Nova::Flavor",
      "updated_at": "2014-08-28T18:13:04Z"
    },
    {
      "created_at": "2014-08-28T18:13:04Z",
      "name": "OS::Nova::Aggregate",
      "updated_at": "2014-08-28T18:13:04Z"
    },
    {
      "created_at": "2014-08-28T18:13:04Z",
      "name": "OS::Nova::Instance",
      "updated_at": "2014-08-28T18:13:04Z"
    }
  ]
}
```


12.7.2. Create resource type association

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace_id}/resource_types	Creates a resource type association in a namespace.

Normal response codes: 201

12.7.2.1. Request

This table shows the URI parameters for the create resource type association request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

Example 12.18. Create resource type association: JSON request

```
{
  "name": "OS::Cinder::Volume",
  "prefix": "hw_",
  "properties_target": "image"
}
```

12.7.2.2. Response

Example 12.19. Create resource type association: JSON response

```
{
  "created_at": "2014-09-19T16:09:13Z",
  "name": "OS::Cinder::Volume",
  "prefix": "hw_",
  "properties_target": "image",
  "updated_at": "2014-09-19T16:09:13Z"
}
```

12.7.3. List resource type associations

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace_id}/resource_types	Lists resource type associations in a namespace.

The response body lists resource type association entities.

Normal response codes: 200

12.7.3.1. Request

This table shows the URI parameters for the list resource type associations request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

This operation does not accept a request body.

12.7.3.2. Response

Example 12.20. List resource type associations: JSON response

```
{
  "resource_types": [
    {
      "created_at": "2014-08-28T18:13:04Z",
      "name": "OS::Glance::Image",
      "updated_at": "2014-08-28T18:13:04Z"
    },
    {
      "created_at": "2014-08-28T18:13:04Z",
      "name": "OS::Cinder::Volume",
      "updated_at": "2014-08-28T18:13:04Z"
    },
    {
      "created_at": "2014-08-28T18:13:04Z",
      "name": "OS::Nova::Flavor",
      "updated_at": "2014-08-28T18:13:04Z"
    },
    {
      "created_at": "2014-08-28T18:13:04Z",
      "name": "OS::Nova::Aggregate",
      "updated_at": "2014-08-28T18:13:04Z"
    },
    {
      "created_at": "2014-08-28T18:13:04Z",
      "name": "OS::Nova::Instance",
      "updated_at": "2014-08-28T18:13:04Z"
    }
  ]
}
```

12.7.4. Remove resource type association

Method	URI	Description
DELETE	/v2/metadefs/namespaces/{namespace_id}/resource_types/{name}	Removes a resource type association in a namespace.

To remove an association, first make an update namespace request to set the protected attribute to false (boolean) on the namespace. Then, remove the association. If the operation succeeds, the response returns the HTTP 204 status code.

If you try to remove resource type associations in a namespace with the protected attribute set to true (boolean), the operation fails and the response returns the HTTP 403 error code.

Normal response codes: 204

Error response codes: forbidden (403)

12.7.4.1. Request

This table shows the URI parameters for the remove resource type association request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.
{name}	String	Name of the resource type.

This operation does not accept a request body.

12.8. Metadata definition namespaces (since API v2.0) (metadefs, namespaces)

Creates, lists, shows details for, updates, and deletes metadata definition namespaces. Defines namespaces that can contain property definitions, object definitions, and resource type associations.

Method	URI	Description
POST	/v2/metadefs/namespaces	Creates a namespace.
GET	/v2/metadefs/namespaces{?limit, marker, visibility, resource_types, sort_key, sort_dir}	Lists public namespaces.
GET	/v2/metadefs/namespaces/{namespace_id}	Gets details for a namespace.
PUT	/v2/metadefs/namespaces/{namespace_id}	Updates a namespace.
DELETE	/v2/metadefs/namespaces/{namespace_id}	Deletes a namespace and its properties, objects, and any resource type associations.

12.8.1. Create namespace

Method	URI	Description
POST	/v2/metadefs/namespaces	Creates a namespace.

The Location response header contains the newly-created URI for the namespace.

Normal response codes: 201

12.8.1.1. Request

Example 12.21. Create namespace: JSON request

```
{
  "description": "Choose capabilities that should be provided by the Compute
  Host. This provides the ability to fine tune the hardware specification
  required when a new vm is requested.",
  "display_name": "Hypervisor Selection",
  "namespace": "OS::Compute::Hypervisor",
  "properties": {
    "hypervisor_type": {
      "description": "The hypervisor type.",
      "enum": [
        "xen",
        "qemu",
        "kvm",
        "lxc",
        "uml",
        "vmware",
        "hyperv"
      ],
      "title": "Hypervisor Type",
      "type": "string"
    },
    "vm_mode": {
      "description": "The virtual machine mode.",
      "enum": [
        "hvm",
        "xen",
        "uml",
        "exe"
      ],
      "title": "VM Mode",
      "type": "string"
    }
  },
  "protected": true,
  "resource_type_associations": [
    {
      "name": "OS::Glance::Image"
    }
  ],
  "visibility": "public"
}
```

12.8.1.2. Response

Example 12.22. Create namespace: JSON response

```
{
  "description": "Choose capabilities that should be provided by the Compute
Host. This provides the ability to fine tune the hardware specification
required when a new vm is requested.",
  "display_name": "Hypervisor Selection",
  "namespace": "OS::Compute::Hypervisor",
  "properties": {
    "hypervisor_type": {
      "description": "The hypervisor type.",
      "enum": [
        "xen",
        "qemu",
        "kvm",
        "lxc",
        "uml",
        "vmware",
        "hyperv"
      ],
      "title": "Hypervisor Type",
      "type": "string"
    },
    "vm_mode": {
      "description": "The virtual machine mode.",
      "enum": [
        "hvm",
        "xen",
        "uml",
        "exe"
      ],
      "title": "VM Mode",
      "type": "string"
    }
  },
  "protected": true,
  "resource_type_associations": [
    {
      "name": "OS::Glance::Image"
    }
  ],
  "schema": "/v2/schemas/metadefs/namespace",
  "self": "/v2/metadefs/namespaces/OS::Compute::Hypervisor",
  "visibility": "public"
}
```

12.8.2. List namespaces

Method	URI	Description
GET	/v2/metadefs/namespaces{?limit, marker, visibility, resource_types, sort_key, sort_dir}	Lists public namespaces.

Returns a subset in the larger collection of namespaces and a link that you can use to get the next set of namespaces. Check for the presence of a `next` link and use it as the URI in a subsequent HTTP GET request. Follow this pattern until a `next` link is no longer provided. The `next` link preserves any query parameters that you send in your initial request. You can use the `first` link to return to the first page in the collection. If you prefer to paginate through namespaces manually, use the `limit` and `marker` parameters.

The list operation accepts the `resource_types` and `visibility` query parameters, which you can use to filter the response.

To sort the results of this operation, use the `sort_key` and `sort_dir` parameters. The API uses the natural sorting order in the namespace attribute that you provide as the `sort_key` parameter.

Normal response codes: 200

12.8.2.1. Request

This table shows the query parameters for the list namespaces request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
visibility	String (Optional)	Filters the response by an image visibility value or values. Use the comma (,) character to separate multiple values. A valid value is <code>public</code> , <code>private</code> , or <code>shared</code> . If you omit this parameter, the response shows <code>public</code> , <code>private</code> , and <code>shared</code> images with a member status of <code>accepted</code> .
resource_types	Int (Optional)	Filters the response by a resource type or types. Use the comma (,) character to separate multiple values. For example, <code>OS::Glance::Image</code> , <code>OS::Nova::Flavor</code> shows only namespaces for these resource types.
sort_key	String (Optional)	Sorts the response by an attribute, such as <code>name</code> , <code>id</code> , or <code>updated_at</code> . Default is <code>created_at</code> . The API uses the natural sorting direction of the <code>sort_key</code> image attribute.
sort_dir	String (Optional)	Sorts the response by a set of one or more sort direction and attribute (<code>sort_key</code>) combinations. A valid value for the sort direction is <code>asc</code> (ascending) or <code>desc</code> (descending). If you omit the sort direction in a set, the default is <code>desc</code> .

This operation does not accept a request body.

12.8.2.2. Response

Example 12.23. List namespaces: JSON response

```
{
  "first": "/v2/metadefs/namespaces?sort_key=created_at&sort_dir=asc",
  "namespaces": [
    {
      "created_at": "2014-08-28T17:13:06Z",
      "description": "The libvirt compute driver options. These are
properties specific to compute drivers. For a list of all hypervisors, see
here: https://wiki.openstack.org/wiki/HypervisorSupportMatrix.",
      "display_name": "libvirt Driver Options",
      "namespace": "OS::Compute::Libvirt",
      "owner": "admin",
      "protected": true,
      "resource_type_associations": [
        {
          "created_at": "2014-08-28T17:13:06Z",
          "name": "OS::Glance::Image",
          "updated_at": "2014-08-28T17:13:06Z"
        }
      ],
      "schema": "/v2/schemas/metadefs/namespace",
      "self": "/v2/metadefs/namespaces/OS::Compute::Libvirt",
      "updated_at": "2014-08-28T17:13:06Z",
      "visibility": "public"
    },
    {
      "created_at": "2014-08-28T17:13:06Z",
      "description": "Compute drivers may enable quotas on CPUs
available to a VM, disk tuning, bandwidth I/O, and instance VIF traffic
control. See: http://docs.openstack.org/admin-guide-cloud/compute-flavors.
html",
      "display_name": "Flavor Quota",
      "namespace": "OS::Compute::Quota",
      "owner": "admin",
      "protected": true,
      "resource_type_associations": [
        {
          "created_at": "2014-08-28T17:13:06Z",
          "name": "OS::Nova::Flavor",
          "updated_at": "2014-08-28T17:13:06Z"
        }
      ],
      "schema": "/v2/schemas/metadefs/namespace",
      "self": "/v2/metadefs/namespaces/OS::Compute::Quota",
      "updated_at": "2014-08-28T17:13:06Z",
      "visibility": "public"
    },
    {
      "created_at": "2014-08-28T17:13:06Z",
      "description": "Trusted compute pools with Intel\u00ae Trusted
Execution Technology (Intel\u00ae TXT) support IT compliance by protecting
virtualized data centers - private, public, and hybrid clouds against
attacks toward hypervisor and BIOS, firmware, and other pre-launch software
components.",

```

```

        "display_name": "Trusted Compute Pools (Intel\u00ae TXT)",
        "namespace": "OS::Compute::Trust",
        "owner": "admin",
        "protected": true,
        "resource_type_associations": [
            {
                "created_at": "2014-08-28T17:13:06Z",
                "name": "OS::Nova::Flavor",
                "updated_at": "2014-08-28T17:13:06Z"
            }
        ],
        "schema": "/v2/schemas/metadefs/namespace",
        "self": "/v2/metadefs/namespaces/OS::Compute::Trust",
        "updated_at": "2014-08-28T17:13:06Z",
        "visibility": "public"
    },
    {
        "created_at": "2014-08-28T17:13:06Z",
        "description": "This provides the preferred socket/core/thread
counts for the virtual CPU instance exposed to guests. This enables the
ability to avoid hitting limitations on vCPU topologies that OS vendors place
on their products. See also: http://git.openstack.org/cgit/openstack/nova-specs/tree/specs/juno/virt-driver-vcpu-topology.rst",
        "display_name": "Virtual CPU Topology",
        "namespace": "OS::Compute::VirtCPUTopology",
        "owner": "admin",
        "protected": true,
        "resource_type_associations": [
            {
                "created_at": "2014-08-28T17:13:06Z",
                "name": "OS::Glance::Image",
                "prefix": "hw_",
                "updated_at": "2014-08-28T17:13:06Z"
            },
            {
                "created_at": "2014-08-28T17:13:06Z",
                "name": "OS::Cinder::Volume",
                "prefix": "hw_",
                "properties_target": "image",
                "updated_at": "2014-08-28T17:13:06Z"
            },
            {
                "created_at": "2014-08-28T17:13:06Z",
                "name": "OS::Nova::Flavor",
                "prefix": "hw:",
                "updated_at": "2014-08-28T17:13:06Z"
            }
        ],
        "schema": "/v2/schemas/metadefs/namespace",
        "self": "/v2/metadefs/namespaces/OS::Compute::VirtCPUTopology",
        "updated_at": "2014-08-28T17:13:06Z",
        "visibility": "public"
    }
],
"schema": "/v2/schemas/metadefs/namespaces"
}

```


12.8.3. Get namespaces details

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace_id}	Gets details for a namespace.

The response body shows a single namespace entity with all details including properties and objects.

Normal response codes: 200

12.8.3.1. Request

This table shows the URI parameters for the get namespaces details request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

This operation does not accept a request body.

12.8.3.2. Response

Example 12.24. Get namespaces details: JSON response

```
{
  "created_at": "2014-08-28T17:13:06Z",
  "description": "The libvirt compute driver options. These are properties specific to compute drivers. For a list of all hypervisors, see here: https://wiki.openstack.org/wiki/HypervisorSupportMatrix.",
  "display_name": "libvirt Driver Options",
  "namespace": "OS::Compute::Libvirt",
  "owner": "admin",
  "properties": {
    "hw_disk_bus": {
      "description": "Specifies the type of disk controller to attach disk devices to.",
      "enum": [
        "scsi",
        "virtio",
        "uml",
        "xen",
        "ide",
        "usb"
      ],
      "title": "Disk Bus",
      "type": "string"
    },
    "hw_machine_type": {
      "description": "Enables booting an ARM system using the specified machine type. By default, if an ARM image is used and its type is not specified, Compute uses vexpress-a15 (for ARMv7) or virt (for AArch64) machine types. Valid types can be viewed by using the virsh capabilities command (machine types are displayed in the machine tag).",
      "title": "Machine Type",
      "type": "string"
    },
    "hw_qemu_guest_agent": {
```

```

        "description": "It is a daemon program running inside the domain
        which is supposed to help management applications with executing functions
        which need assistance of the guest OS. For example, freezing and thawing
        filesystems, entering suspend. However, guest agent (GA) is not bullet proof,
        and hostile guest OS can send spurious replies.",
        "enum": [
            "yes",
            "no"
        ],
        "title": "QEMU Guest Agent",
        "type": "string"
    },
    "hw_rng_model": {
        "default": "virtio",
        "description": "Adds a random-number generator device to the
        image's instances. The cloud administrator can enable and control device
        behavior by configuring the instance's flavor. By default: The generator
        device is disabled. /dev/random is used as the default entropy source. To
        specify a physical HW RNG device, use the following option in the nova.conf
        file: rng_dev_path=/dev/hwrng",
        "title": "Random Number Generator Device",
        "type": "string"
    },
    "hw_scsi_model": {
        "default": "virtio-scsi",
        "description": "Enables the use of VirtIO SCSI (virtio-scsi) to
        provide block device access for compute instances; by default, instances use
        VirtIO Block (virtio-blk). VirtIO SCSI is a para-virtualized SCSI controller
        device that provides improved scalability and performance, and supports
        advanced SCSI hardware.",
        "title": "SCSI Model",
        "type": "string"
    },
    "hw_video_model": {
        "description": "The video image driver used.",
        "enum": [
            "vga",
            "cirrus",
            "vmvga",
            "xen",
            "qxl"
        ],
        "title": "Video Model",
        "type": "string"
    },
    "hw_video_ram": {
        "description": "Maximum RAM for the video image. Used only if a
        hw_video:ram_max_mb value has been set in the flavor's extra_specs and that
        value is higher than the value set in hw_video_ram.",
        "title": "Max Video Ram",
        "type": "integer"
    },
    "hw_vif_model": {
        "description": "Specifies the model of virtual network interface
        device to use. The valid options depend on the configured hypervisor. KVM
        and QEMU: e1000, ne2k_pci, pcnet, rtl8139, and virtio. VMware: e1000, e1000e,
        VirtualE1000, VirtualE1000e, VirtualPCNet32, VirtualSriovEthernetCard, and
        VirtualVmxnet. Xen: e1000, netfront, ne2k_pci, pcnet, and rtl8139.",
        "enum": [
            "e1000",

```

```

        "ne2k_pci",
        "pcnet",
        "rtl8139",
        "virtio",
        "e1000",
        "e1000e",
        "VirtualE1000",
        "VirtualE1000e",
        "VirtualPCNet32",
        "VirtualSriovEthernetCard",
        "VirtualVmxnet",
        "netfront",
        "ne2k_pci"
    ],
    "title": "Virtual Network Interface",
    "type": "string"
},
"os_command_line": {
    "description": "The kernel command line to be used by the libvirt
driver, instead of the default. For linux containers (LXC), the value is used
as arguments for initialization. This key is valid only for Amazon kernel,
ramdisk, or machine images (aki, ari, or ami).",
    "title": "Kernel Command Line",
    "type": "string"
}
},
"protected": true,
"resource_type_associations": [
    {
        "created_at": "2014-08-28T17:13:06Z",
        "name": "OS::Glance::Image",
        "updated_at": "2014-08-28T17:13:06Z"
    }
],
"schema": "/v2/schemas/metadefs/namespace",
"self": "/v2/metadefs/namespaces/OS::Compute::Libvirt",
"updated_at": "2014-08-28T17:13:06Z",
"visibility": "public"
}

```

12.8.4. Update namespace

Method	URI	Description
PUT	/v2/metadefs/namespaces/{namespace_id}	Updates a namespace.

Normal response codes: 200

12.8.4.1. Request

This table shows the URI parameters for the update namespace request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

Example 12.25. Update namespace: JSON request

```
{
  "description": "Choose capabilities that should be provided by the Compute
Host. This provides the ability to fine tune the hardware specification
required when a new vm is requested.",
  "display_name": "Hypervisor Selection",
  "namespace": "OS::Compute::Hypervisor",
  "protected": false,
  "visibility": "public"
}
```

12.8.4.2. Response

Example 12.26. Update namespace: JSON response

```
{
  "created_at": "2014-09-19T13:31:37Z",
  "description": "Choose capabilities that should be provided by the Compute
Host. This provides the ability to fine tune the hardware specification
required when a new vm is requested.",
  "display_name": "Hypervisor Selection",
  "namespace": "OS::Compute::Hypervisor",
  "owner": "7ec22942411e427692e8a3436be1031a",
  "protected": false,
  "schema": "/v2/schemas/metadefs/namespace",
  "self": "/v2/metadefs/namespaces/OS::Compute::Hypervisor",
  "updated_at": "2014-09-19T13:31:37Z",
  "visibility": "public"
}
```

12.8.5. Delete namespace

Method	URI	Description
DELETE	/v2/metadefs/namespaces/{namespace_id}	Deletes a namespace and its properties, objects, and any resource type associations.

You cannot delete namespaces with the `protected` attribute set to `true` (boolean); the response returns the HTTP 403 response code.

To delete a namespace, you must first make an update namespace request to set the `protected` attribute to `false` (boolean) on the namespace. Then, delete the namespace.

A successful operation returns the HTTP 204 response code.

If you try to remove a namespace with the `protected` attribute set to `true` (boolean), the operation fails and the response returns the HTTP 403 response code.

Normal response codes: 204

Error response codes: forbidden (403)

12.8.5.1. Request

This table shows the URI parameters for the delete namespace request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

This operation does not accept a request body.

12.9. Metadata definition properties (since API v2.0) (metadefs, namespaces, properties)

Creates, lists, shows details for, updates, and deletes metadata definition properties.

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace_id}/properties	Creates a property definition in a namespace.
GET	/v2/metadefs/namespaces/{namespace_id}/properties	Lists property definitions in a namespace.
GET	/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}{?resource_type}	Shows the definition for a property.
PUT	/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}	Updates a property definition.
DELETE	/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}	Removes a property definition in a namespace.

12.9.1. Create property

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace_id}/properties	Creates a property definition in a namespace.

The schema is a subset of the JSON property definition schema.

Normal response codes: 201

12.9.1.1. Request

This table shows the URI parameters for the create property request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

Example 12.27. Create property: JSON request

```
{
  "description": "The hypervisor type. It may be used by the host properties
filter for scheduling. The ImagePropertiesFilter filters compute nodes that
satisfy any architecture, hypervisor type, or virtual machine mode properties
specified on the instance's image properties. Image properties are contained
in the image dictionary in the request_spec.",
  "enum": [
    "xen",
    "qemu",
    "kvm",
    "lxc",
    "uml",
    "vmware",
    "hyperv"
  ],
  "name": "hypervisor_type",
  "title": "Hypervisor Type",
  "type": "string"
}
```

12.9.1.2. Response

Example 12.28. Create property: JSON response

```
{
  "description": "The hypervisor type. It may be used by the host properties
filter for scheduling. The ImagePropertiesFilter filters compute nodes that
satisfy any architecture, hypervisor type, or virtual machine mode properties
specified on the instance's image properties. Image properties are contained
in the image dictionary in the request_spec.",
  "enum": [
    "xen",
    "qemu",
    "kvm",
    "lxc",
    "uml",

```

```
        "vmware",  
        "hyperv"  
    ],  
    "name": "hypervisor_type",  
    "title": "Hypervisor Type",  
    "type": "string"  
}
```

12.9.2. List property definitions

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace_id}/properties	Lists property definitions in a namespace.

Normal response codes: 200

12.9.2.1. Request

This table shows the URI parameters for the list property definitions request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

This operation does not accept a request body.

12.9.2.2. Response

Example 12.29. List property definitions: JSON response

```
{
  "properties": {
    "hw_disk_bus": {
      "description": "Specifies the type of disk controller to attach
disk devices to.",
      "enum": [
        "scsi",
        "virtio",
        "uml",
        "xen",
        "ide",
        "usb"
      ],
      "title": "Disk Bus",
      "type": "string"
    },
    "hw_machine_type": {
      "description": "Enables booting an ARM system using the specified
machine type. By default, if an ARM image is used and its type is not
specified, Compute uses vexpress-a15 (for ARMv7) or virt (for AArch64)
machine types. Valid types can be viewed by using the virsh capabilities
command (machine types are displayed in the machine tag).",
      "title": "Machine Type",
      "type": "string"
    },
    "hw_qemu_guest_agent": {
      "description": "It is a daemon program running inside the domain
which is supposed to help management applications with executing functions
which need assistance of the guest OS. For example, freezing and thawing
filesystems, entering suspend. However, guest agent (GA) is not bullet proof,
and hostile guest OS can send spurious replies.",
      "enum": [
        "yes",
        "no"
      ],
      "title": "QEMU Guest Agent",
```



```

        "type": "string"
    },
    "hw_rng_model": {
        "default": "virtio",
        "description": "Adds a random-number generator device to the
image's instances. The cloud administrator can enable and control device
behavior by configuring the instance's flavor. By default: The generator
device is disabled. /dev/random is used as the default entropy source. To
specify a physical HW RNG device, use the following option in the nova.conf
file: rng_dev_path=/dev/hwrng",
        "title": "Random Number Generator Device",
        "type": "string"
    },
    "hw_scsi_model": {
        "default": "virtio-scsi",
        "description": "Enables the use of VirtIO SCSI (virtio-scsi) to
provide block device access for compute instances; by default, instances use
VirtIO Block (virtio-blk). VirtIO SCSI is a para-virtualized SCSI controller
device that provides improved scalability and performance, and supports
advanced SCSI hardware.",
        "title": "SCSI Model",
        "type": "string"
    },
    "hw_video_model": {
        "description": "The video image driver used.",
        "enum": [
            "vga",
            "cirrus",
            "vmvga",
            "xen",
            "qxl"
        ],
        "title": "Video Model",
        "type": "string"
    },
    "hw_video_ram": {
        "description": "Maximum RAM for the video image. Used only if a
hw_video:ram_max_mb value has been set in the flavor's extra_specs and that
value is higher than the value set in hw_video_ram.",
        "title": "Max Video Ram",
        "type": "integer"
    },
    "hw_vif_model": {
        "description": "Specifies the model of virtual network interface
device to use. The valid options depend on the configured hypervisor. KVM
and QEMU: e1000, ne2k_pci, pcnet, rtl8139, and virtio. VMware: e1000, e1000e,
VirtualE1000, VirtualE1000e, VirtualPCNet32, VirtualSriovEthernetCard, and
VirtualVmxnet. Xen: e1000, netfront, ne2k_pci, pcnet, and rtl8139.",
        "enum": [
            "e1000",
            "ne2k_pci",
            "pcnet",
            "rtl8139",
            "virtio",
            "e1000",
            "e1000e",
            "VirtualE1000",
            "VirtualE1000e",
            "VirtualPCNet32",
            "VirtualSriovEthernetCard",

```

```
        "VirtualVmxnet",
        "netfront",
        "ne2k_pci"
    ],
    "title": "Virtual Network Interface",
    "type": "string"
},
"os_command_line": {
    "description": "The kernel command line to be used by the libvirt
driver, instead of the default. For linux containers (LXC), the value is used
as arguments for initialization. This key is valid only for Amazon kernel,
ramdisk, or machine images (aki, ari, or ami).",
    "title": "Kernel Command Line",
    "type": "string"
}
}
```

12.9.3. Show property definition

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}{?resource_type}	Shows the definition for a property.

If you use the `resource_type` query parameter, the API removes the prefix of the resource type from the property name before it submits the query. This enables you to look for a property name that starts with a prefix from an associated resource type.

The response body shows a single property entity.

Normal response codes: 200

12.9.3.1. Request

This table shows the URI parameters for the show property definition request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.
{property_name}	String	The name of the property.

This table shows the query parameters for the show property definition request:

Name	Type	Description
resource_type	Int (Optional)	Filters the response by property names that start with a prefix from an associated resource type. The API removes the prefix of the resource type from the property name in the response.

This operation does not accept a request body.

12.9.3.2. Response

Example 12.30. Show property definition: JSON response

```
{
  "description": "The hypervisor type. It may be used by the host properties filter for scheduling. The ImagePropertiesFilter filters compute nodes that satisfy any architecture, hypervisor type, or virtual machine mode properties specified on the instance's image properties. Image properties are contained in the image dictionary in the request_spec.",
  "enum": [
    "xen",
    "qemu",
    "kvm",
    "lxc",
    "uml",
    "vmware",
    "hyperv"
  ],
}
```

```
    "name": "hypervisor_type",  
    "title": "Hypervisor Type",  
    "type": "string"  
}
```

12.9.4. Update property definition

Method	URI	Description
PUT	/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}	Updates a property definition.

Normal response codes: 200

12.9.4.1. Request

This table shows the URI parameters for the update property definition request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.
{property_name}	String	The name of the property.

Example 12.31. Update property definition: JSON request

```
{
  "description": "The hypervisor type. It may be used by the host properties
filter for scheduling. The ImagePropertiesFilter filters compute nodes that
satisfy any architecture, hypervisor type, or virtual machine mode properties
specified on the instance's image properties. Image properties are contained
in the image dictionary in the request_spec.",
  "enum": [
    "xen",
    "qemu",
    "kvm",
    "lxc",
    "uml",
    "vmware",
    "hyperv"
  ],
  "name": "hypervisor_type",
  "title": "Hypervisor Type",
  "type": "string"
}
```

12.9.4.2. Response

Example 12.32. Update property definition: JSON response

```
{
  "description": "The hypervisor type. It may be used by the host properties
filter for scheduling. The ImagePropertiesFilter filters compute nodes that
satisfy any architecture, hypervisor type, or virtual machine mode properties
specified on the instance's image properties. Image properties are contained
in the image dictionary in the request_spec.",
  "enum": [
    "xen",
    "qemu",
    "kvm",
    "lxc",
    "uml",
```

```
        "vmware",  
        "hyperv"  
    ],  
    "name": "hypervisor_type",  
    "title": "Hypervisor Type",  
    "type": "string"  
}
```

12.9.5. Remove property definition

Method	URI	Description
DELETE	/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}	Removes a property definition in a namespace.

To remove a property, first make an update namespace request to set the protected attribute to false (boolean) on the namespace. Then, remove the property. If the operation succeeds, the response returns the HTTP 204 status code.

If you try to remove a property in a namespace with the protected attribute set to true (boolean), the operation fails and the response returns the HTTP 403 error code.

Normal response codes: 204

Error response codes: forbidden (403)

12.9.5.1. Request

This table shows the URI parameters for the remove property definition request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.
{property_name}	String	The name of the property.

This operation does not accept a request body.

12.10. Metadata definition objects (since API v2.0) (metadefs, namespaces, objects)

Creates, lists, shows details for, updates, and deletes metadata definition objects.

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace_id}/objects	Creates an object definition in a namespace.
GET	/v2/metadefs/namespaces/{namespace_id}/objects{?visibility,resource_types,sort_key,sort_dir}	Lists object definitions in a namespace.
GET	/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}	Shows the definition for an object.
PUT	/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}	Updates an object definition in a namespace.
DELETE	/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}	Deletes an object definition from a namespace.

12.10.1. Create object

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace_id}/objects	Creates an object definition in a namespace.

Normal response codes: 201

12.10.1.1. Request

This table shows the URI parameters for the create object request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

Example 12.33. Create object: JSON request

```
{
  "description": "You can configure the CPU limits with control parameters.",
  "name": "CPU Limits",
  "properties": {
    "quota:cpu_period": {
      "description": "Specifies the enforcement interval (unit:
microseconds) for QEMU and LXC hypervisors. Within a period, each VCPU of
the domain is not allowed to consume more than the quota worth of runtime.
The value should be in range [1000, 1000000]. A period with value 0 means no
value.",
      "maximum": 1000000,
      "minimum": 1000,
      "title": "Quota: CPU Period",
      "type": "integer"
    },
    "quota:cpu_quota": {
      "description": "Specifies the maximum allowed bandwidth (unit:
microseconds). A domain with a negative-value quota indicates that the domain
has infinite bandwidth, which means that it is not bandwidth controlled. The
value should be in range [1000, 18446744073709551] or less than 0. A quota
with value 0 means no value. You can use this feature to ensure that all
vCPUs run at the same speed.",
      "title": "Quota: CPU Quota",
      "type": "integer"
    },
    "quota:cpu_shares": {
      "description": "Specifies the proportional weighted share for the
domain. If this element is omitted, the service defaults to the OS provided
defaults. There is no unit for the value; it is a relative measure based on
the setting of other VMs. For example, a VM configured with value 2048 gets
twice as much CPU time as a VM configured with value 1024.",
      "title": "Quota: CPU Shares",
      "type": "integer"
    }
  },
  "required": []
}
```


12.10.1.2. Response

Example 12.34. Create object: JSON response

```
{
  "created_at": "2014-09-19T18:20:56Z",
  "description": "You can configure the CPU limits with control parameters.",
  "name": "CPU Limits",
  "properties": {
    "quota:cpu_period": {
      "description": "Specifies the enforcement interval (unit:
microseconds) for QEMU and LXC hypervisors. Within a period, each VCPU of
the domain is not allowed to consume more than the quota worth of runtime.
The value should be in range [1000, 1000000]. A period with value 0 means no
value.",
      "maximum": 1000000,
      "minimum": 1000,
      "title": "Quota: CPU Period",
      "type": "integer"
    },
    "quota:cpu_quota": {
      "description": "Specifies the maximum allowed bandwidth (unit:
microseconds). A domain with a negative-value quota indicates that the domain
has infinite bandwidth, which means that it is not bandwidth controlled. The
value should be in range [1000, 18446744073709551] or less than 0. A quota
with value 0 means no value. You can use this feature to ensure that all
vCPUs run at the same speed.",
      "title": "Quota: CPU Quota",
      "type": "integer"
    },
    "quota:cpu_shares": {
      "description": "Specifies the proportional weighted share for the
domain. If this element is omitted, the service defaults to the OS provided
defaults. There is no unit for the value; it is a relative measure based on
the setting of other VMs. For example, a VM configured with value 2048 gets
twice as much CPU time as a VM configured with value 1024.",
      "title": "Quota: CPU Shares",
      "type": "integer"
    }
  },
  "required": [],
  "schema": "/v2/schemas/metadefs/object",
  "self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/CPU Limits",
  "updated_at": "2014-09-19T18:20:56Z"
}
```

12.10.2. List object definitions

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace_id}/objects{?visibility,resource_types,sort_key,sort_dir}	Lists object definitions in a namespace.

Returns a subset of the larger collection of namespaces and a link that you can use to get the next set of namespaces. You should always check for the presence of a next link and use it as the URI in a subsequent HTTP GET request. You should follow this pattern until a next link is no longer provided. The next link preserves any query parameters that you send in your initial request. You can use the first link to jump back to the first page of the collection. If you prefer to paginate through namespaces manually, use the `limit` and `marker` parameters.

Use the `resource_types` and `visibility` query parameters to filter the response.

For example, set the `resource_types` query parameter to `OS::Glance::Image,OS::Nova::Flavor` to filter the response to include only namespaces that are associated with the given resource types.

You can sort the results of this operation by using the `sort_key` and `sort_dir` parameters. The API uses the natural sorting of whatever namespace attribute is provided as the `sort_key`.

Normal response codes: 200

12.10.2.1. Request

This table shows the URI parameters for the list object definitions request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

This table shows the query parameters for the list object definitions request:

Name	Type	Description
visibility	String (Optional)	Filters the response by a namespace visibility value. A valid value is <code>public</code> or <code>private</code> . If you omit this parameter, the response shows <code>public</code> and <code>private</code> namespaces.
resource_types	Int (Optional)	Filters the response by a resource type or types. Use the comma (,) character to separate multiple values. For example, <code>OS::Glance::Image,OS::Nova::Flavor</code> shows only namespaces for these resource types.
sort_key	String (Optional)	Sorts the response by an attribute, such as <code>name</code> , <code>id</code> , or <code>updated_at</code> . Default is <code>created_at</code> . The API uses the natural sorting direction of the <code>sort_key</code> image attribute.
sort_dir	String (Optional)	Sorts the response by a set of one or more sort direction and attribute (<code>sort_key</code>) combinations. A valid value for the sort direction is <code>asc</code> (ascending) or <code>desc</code> (descending). If you omit the sort direction in a set, the default is <code>desc</code> .

This operation does not accept a request body.

12.10.2.2. Response

Example 12.35. List object definitions: JSON response

```
{
  "objects": [
    {
      "created_at": "2014-09-18T18:16:35Z",
      "description": "You can configure the CPU limits with control
parameters.",
      "name": "CPU Limits",
      "properties": {
        "quota:cpu_period": {
          "description": "Specifies the enforcement interval (unit:
microseconds) for QEMU and LXC hypervisors. Within a period, each VCPU of
the domain is not allowed to consume more than the quota worth of runtime.
The value should be in range [1000, 1000000]. A period with value 0 means no
value.",
          "maximum": 1000000,
          "minimum": 1000,
          "title": "Quota: CPU Period",
          "type": "integer"
        },
        "quota:cpu_quota": {
          "description": "Specifies the maximum allowed bandwidth
(unit: microseconds). A domain with a negative-value quota indicates that
the domain has infinite bandwidth, which means that it is not bandwidth
controlled. The value should be in range [1000, 18446744073709551] or less
than 0. A quota with value 0 means no value. You can use this feature to
ensure that all vCPUs run at the same speed.",
          "title": "Quota: CPU Quota",
          "type": "integer"
        },
        "quota:cpu_shares": {
          "description": "Specifies the proportional weighted share
for the domain. If this element is omitted, the service defaults to the OS
provided defaults. There is no unit for the value; it is a relative measure
based on the setting of other VMs. For example, a VM configured with value
2048 gets twice as much CPU time as a VM configured with value 1024.",
          "title": "Quota: CPU Shares",
          "type": "integer"
        }
      },
      "required": [],
      "schema": "/v2/schemas/metadefs/object",
      "self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/CPU
Limits"
    },
    {
      "created_at": "2014-09-18T18:16:35Z",
      "description": "Using disk I/O quotas, you can set maximum disk
write to 10 MB per second for a VM user.",
      "name": "Disk QoS",
      "properties": {
        "quota:disk_read_bytes_sec": {
          "description": "Sets disk I/O quota for disk read bytes /
sec.",
          "title": "Quota: Disk read bytes / sec",
          "type": "integer"
        }
      }
    }
  ]
}
```

```

    },
    "quota:disk_read_iops_sec": {
      "description": "Sets disk I/O quota for disk read IOPS /
sec.",
      "title": "Quota: Disk read IOPS / sec",
      "type": "integer"
    },
    "quota:disk_total_bytes_sec": {
      "description": "Sets disk I/O quota for total disk bytes /
sec.",
      "title": "Quota: Disk Total Bytes / sec",
      "type": "integer"
    },
    "quota:disk_total_iops_sec": {
      "description": "Sets disk I/O quota for disk total IOPS /
sec.",
      "title": "Quota: Disk Total IOPS / sec",
      "type": "integer"
    },
    "quota:disk_write_bytes_sec": {
      "description": "Sets disk I/O quota for disk write bytes /
sec.",
      "title": "Quota: Disk Write Bytes / sec",
      "type": "integer"
    },
    "quota:disk_write_iops_sec": {
      "description": "Sets disk I/O quota for disk write IOPS /
sec.",
      "title": "Quota: Disk Write IOPS / sec",
      "type": "integer"
    }
  },
  "required": [],
  "schema": "/v2/schemas/metadefs/object",
  "self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/Disk
QoS"
},
{
  "created_at": "2014-09-18T18:16:35Z",
  "description": "Bandwidth QoS tuning for instance virtual
interfaces (VIFs) may be specified with these properties. Incoming and
outgoing traffic can be shaped independently. If not specified, no quality
of service (QoS) is applied on that traffic direction. So, if you want to
shape only the network's incoming traffic, use inbound only (and vice versa).
The OpenStack Networking service abstracts the physical implementation of
the network, allowing plugins to configure and manage physical resources.
Virtual Interfaces (VIF) in the logical model are analogous to physical
network interface cards (NICs). VIFs are typically owned a managed by an
external service; for instance when OpenStack Networking is used for building
OpenStack networks, VIFs would be created, owned, and managed in Nova. VIFs
are connected to OpenStack Networking networks via ports. A port is analogous
to a port on a network switch, and it has an administrative state. When a
VIF is attached to a port the OpenStack Networking API creates an attachment
object, which specifies the fact that a VIF with a given identifier is
plugged into the port.",
  "name": "Virtual Interface QoS",
  "properties": {
    "quota:vif_inbound_average": {

```

```

        "description": "Network Virtual Interface (VIF) inbound
average in kilobytes per second. Specifies average bit rate on the interface
being shaped.",
        "title": "Quota: VIF Inbound Average",
        "type": "integer"
    },
    "quota:vif_inbound_burst": {
        "description": "Network Virtual Interface (VIF) inbound
burst in total kilobytes. Specifies the amount of bytes that can be burst at
peak speed.",
        "title": "Quota: VIF Inbound Burst",
        "type": "integer"
    },
    "quota:vif_inbound_peak": {
        "description": "Network Virtual Interface (VIF) inbound
peak in kilobytes per second. Specifies maximum rate at which an interface
can receive data.",
        "title": "Quota: VIF Inbound Peak",
        "type": "integer"
    },
    "quota:vif_outbound_average": {
        "description": "Network Virtual Interface (VIF) outbound
average in kilobytes per second. Specifies average bit rate on the interface
being shaped.",
        "title": "Quota: VIF Outbound Average",
        "type": "integer"
    },
    "quota:vif_outbound_burst": {
        "description": "Network Virtual Interface (VIF) outbound
burst in total kilobytes. Specifies the amount of bytes that can be burst at
peak speed.",
        "title": "Quota: VIF Outbound Burst",
        "type": "integer"
    },
    "quota:vif_outbound_peak": {
        "description": "Network Virtual Interface (VIF) outbound
peak in kilobytes per second. Specifies maximum rate at which an interface
can send data.",
        "title": "Quota: VIF Outbound Burst",
        "type": "integer"
    }
},
"required": [],
"schema": "/v2/schemas/metadefs/object",
"self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/
Virtual Interface QoS"
}
],
"schema": "v2/schemas/metadefs/objects"
}

```

12.10.3. Show object definition

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}	Shows the definition for an object.

The response body shows a single object entity.

Normal response codes: 200

12.10.3.1. Request

This table shows the URI parameters for the show object definition request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.
{object_name}	String	The name of the object.

This operation does not accept a request body.

12.10.3.2. Response

Example 12.36. Show object definition: JSON response

```
{
  "created_at": "2014-09-19T18:20:56Z",
  "description": "You can configure the CPU limits with control parameters.",
  "name": "CPU Limits",
  "properties": {
    "quota:cpu_period": {
      "description": "Specifies the enforcement interval (unit:
microseconds) for QEMU and LXC hypervisors. Within a period, each VCPU of
the domain is not allowed to consume more than the quota worth of runtime.
The value should be in range [1000, 1000000]. A period with value 0 means no
value.",
      "maximum": 1000000,
      "minimum": 1000,
      "title": "Quota: CPU Period",
      "type": "integer"
    },
    "quota:cpu_quota": {
      "description": "Specifies the maximum allowed bandwidth (unit:
microseconds). A domain with a negative-value quota indicates that the domain
has infinite bandwidth, which means that it is not bandwidth controlled. The
value should be in range [1000, 18446744073709551] or less than 0. A quota
with value 0 means no value. You can use this feature to ensure that all
vCPUs run at the same speed.",
      "title": "Quota: CPU Quota",
      "type": "integer"
    },
    "quota:cpu_shares": {
      "description": "Specifies the proportional weighted share for the
domain. If this element is omitted, the service defaults to the OS provided
```

```
defaults. There is no unit for the value; it is a relative measure based on
the setting of other VMs. For example, a VM configured with value 2048 gets
twice as much CPU time as a VM configured with value 1024.",
    "title": "Quota: CPU Shares",
    "type": "integer"
  }
},
"required": [],
"schema": "/v2/schemas/metadefs/object",
"self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/CPU Limits",
"updated_at": "2014-09-19T18:20:56Z"
}
```

12.10.4. Update object definition

Method	URI	Description
PUT	/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}	Updates an object definition in a namespace.

Normal response codes: 200

12.10.4.1. Request

This table shows the URI parameters for the update object definition request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.
{object_name}	String	The name of the object.

Example 12.37. Update object definition: JSON request

```
{
  "description": "You can configure the CPU limits with control parameters.",
  "name": "CPU Limits",
  "properties": {
    "quota:cpu_shares": {
      "description": "Specifies the proportional weighted share for the domain. If this element is omitted, the service defaults to the OS provided defaults. There is no unit for the value; it is a relative measure based on the setting of other VMs. For example, a VM configured with value 2048 gets twice as much CPU time as a VM configured with value 1024.",
      "title": "Quota: CPU Shares",
      "type": "integer"
    }
  },
  "required": []
}
```

12.10.4.2. Response

Example 12.38. Update object definition: JSON response

```
{
  "created_at": "2014-09-19T19:20:56Z",
  "description": "You can configure the CPU limits with control parameters.",
  "name": "CPU Limits",
  "properties": {
    "quota:cpu_shares": {
      "description": "Specifies the proportional weighted share for the domain. If this element is omitted, the service defaults to the OS provided defaults. There is no unit for the value; it is a relative measure based on the setting of other VMs. For example, a VM configured with value 2048 gets twice as much CPU time as a VM configured with value 1024.",
      "title": "Quota: CPU Shares",
      "type": "integer"
    }
  }
}
```



```
    }  
  },  
  "required": [],  
  "schema": "/v2/schemas/metadevs/object",  
  "self": "/v2/metadevs/namespaces/OS::Compute::Quota/objects/CPU Limits",  
  "updated_at": "2014-09-19T19:20:56Z"  
}
```

12.10.5. Delete property definition

Method	URI	Description
DELETE	/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}	Deletes an object definition from a namespace.

To delete a protected object from a namespace, you must first set the protected attribute to false (boolean) on the namespace and then perform the delete. If you try to delete a protected object, the call returns the 403 response code.

When you successfully delete an object from a namespace, the response is empty and the response code is 204.

Normal response codes: 204

Error response codes: forbidden (403)

12.10.5.1. Request

This table shows the URI parameters for the delete property definition request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.
{object_name}	String	The name of the object.

This operation does not accept a request body.

12.11. Metadata definition tags (since API v2.0) (metadefs, namespaces, tags)

Creates, lists, shows details for, updates, and deletes metadata definition tags.

Method	URI	Description
POST	/v2/metadefs/namespaces/tags/{namespace_id}	Creates one or more tag definitions in a namespace.
GET	/v2/metadefs/namespaces/tags/{namespace_id}{?limit,marker,sort_key,sort_dir}	Lists the tag definitions within a namespace.
DELETE	/v2/metadefs/namespaces/tags/{namespace_id}	Deletes all tag definitions within a namespace.
POST	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Adds a tag to the list of namespace tag definitions.
GET	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Gets a definition for a tag.
PUT	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Renames a tag definition.
DELETE	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Deletes a tag definition within a namespace.

12.11.1. Create tags

Method	URI	Description
POST	/v2/metadefs/namespaces/tags/{namespace_id}	Creates one or more tag definitions in a namespace.

Normal response codes: 201

12.11.1.1. Request

This table shows the URI parameters for the create tags request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

This operation does not accept a request body.

12.11.1.2. Response

Example 12.39. Create tags: JSON response

```
{
  "tags": [
    {
      "name": "sample-tag1"
    },
    {
      "name": "sample-tag2"
    },
    {
      "name": "sample-tag3"
    }
  ]
}
```

12.11.2. List tags

Method	URI	Description
GET	/v2/metadefs/namespaces/tags/{namespace_id}{?limit,marker,sort_key,sort_dir}	Lists the tag definitions within a namespace.

To manually paginate through the list of tags, use the `limit` and `marker` parameters.

To sort the results of this operation use the `sort_key` and `sort_dir` parameters. The API uses the natural sort order of the tag attribute of the `sort_key` parameter.

Normal response codes: 200

12.11.2.1. Request

This table shows the URI parameters for the list tags request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

This table shows the query parameters for the list tags request:

Name	Type	Description
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.
sort_key	String (Optional)	Sorts the response by an attribute, such as <code>name</code> , <code>id</code> , or <code>updated_at</code> . Default is <code>created_at</code> . The API uses the natural sorting direction of the <code>sort_key</code> image attribute.
sort_dir	String (Optional)	Sorts the response by a set of one or more sort direction and attribute (<code>sort_key</code>) combinations. A valid value for the sort direction is <code>asc</code> (ascending) or <code>desc</code> (descending). If you omit the sort direction in a set, the default is <code>desc</code> .

This operation does not accept a request body.

12.11.2.2. Response

Example 12.40. List tags: JSON response

```
{
  "tags": [
    {
      "name": "sample-tag1"
    },
    {
      "name": "sample-tag2"
    }
  ],
}
```

```
    {  
      "name": "sample-tag3"  
    }  
  ]  
}
```

12.11.3. Delete all tag definitions

Method	URI	Description
DELETE	/v2/metadefs/namespaces/tags/{namespace_id}	Deletes all tag definitions within a namespace.

You cannot delete tags in a namespace with the 'protected' attribute set to true (boolean); the response returns the HTTP 403 status code.

You must first set the protected attribute to false (boolean) on the namespace and then perform the delete. The response is empty and returns the HTTP 204 status code.

Normal response codes: 204

Error response codes: forbidden (403)

12.11.3.1. Request

This table shows the URI parameters for the delete all tag definitions request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.

This operation does not accept a request body.

12.11.4. Add tag definition

Method	URI	Description
POST	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Adds a tag to the list of namespace tag definitions.

Normal response codes: 200

12.11.4.1. Request

This table shows the URI parameters for the add tag definition request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.
{tag_name}	String	The name of the tag.

This operation does not accept a request body.

12.11.4.2. Response

Example 12.41. Add tag definition: JSON response

```
{
  "created_at": "2015-05-09T01:12:31Z",
  "name": "added-sample-tag",
  "updated_at": "2015-05-09T01:12:31Z"
}
```

12.11.5. Get tag definition

Method	URI	Description
GET	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Gets a definition for a tag.

The response body shows a single tag entity.

Normal response codes: 200

12.11.5.1. Request

This table shows the URI parameters for the get tag definition request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.
{tag_name}	String	The name of the tag.

This operation does not accept a request body.

12.11.5.2. Response

Example 12.42. Get tag definition: JSON response

```
{
  "created_at": "2015-05-06T23:16:12Z",
  "name": "sample-tag2",
  "updated_at": "2015-05-06T23:16:12Z"
}
```


12.11.6. Update tag definition

Method	URI	Description
PUT	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Renames a tag definition.

Normal response codes: 200

12.11.6.1. Request

This table shows the URI parameters for the update tag definition request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.
{tag_name}	String	The name of the tag.

This operation does not accept a request body.

12.11.6.2. Response

Example 12.43. Update tag definition: JSON response

```
{
  "name": "new-tag-name"
}
```

12.11.7. Delete tag definition

Method	URI	Description
DELETE	/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}	Deletes a tag definition within a namespace.

You cannot delete tags in a namespace with the 'protected' attribute set to true (boolean); the response returns the HTTP 403 status code.

You must first set the protected attribute to false (boolean) on the namespace and then perform the delete. The response is empty and returns the HTTP 204 status code.

Normal response codes: 204

Error response codes: forbidden (403)

12.11.7.1. Request

This table shows the URI parameters for the delete tag definition request:

Name	Type	Description
{namespace_id}	UUID	The UUID of the namespace.
{tag_name}	String	The name of the tag.

This operation does not accept a request body.

12.12. Metadata definition schemas (schemas, metadefs, namespace, namespaces, object, objects, property, properties, tag, tags, resource_type, resource_types)

Gets a JSON-schema document that represents a metadata definition entity.

Method	URI	Description
GET	/v2/schemas/metadefs/namespace	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace entity.
GET	/v2/schemas/metadefs/namespaces	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespaces entity.
GET	/v2/schemas/metadefs/object	(Since API v2.1) Shows a JSON schema document that represents a metadata definition object entity.
GET	/v2/schemas/metadefs/objects	(Since API v2.1) Shows a JSON schema document that represents a metadata definition objects entity.
GET	/v2/schemas/metadefs/property	(Since API v2.1) Shows a JSON schema document that represents a metadata definition property entity.
GET	/v2/schemas/metadefs/properties	(Since API v2.1) Shows a JSON schema document that represents a metadata definition properties entity.
GET	/v2/schemas/metadefs/tag	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tag entity.

Method	URI	Description
GET	/v2/schemas/metadefs/tags	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tags entity.
GET	/v2/schemas/metadefs/resource_type	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type association entity.
GET	/v2/schemas/metadefs/resource_types	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type associations entity.

12.12.1. Show metadata definition namespace schema

Method	URI	Description
GET	/v2/schemas/metadefs/namespace	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace entity.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

12.12.1.1. Request

This operation does not accept a request body.

12.12.1.2. Response

Example 12.44. Show metadata definition namespace schema: JSON response

```
{
  "additionalProperties": false,
  "definitions": {
    "positiveInteger": {
      "minimum": 0,
      "type": "integer"
    },
    "positiveIntegerDefault0": {
      "allOf": [
        {
          "$ref": "#/definitions/positiveInteger"
        },
        {
          "default": 0
        }
      ]
    },
    "property": {
      "additionalProperties": {
        "properties": {
          "additionalItems": {
            "type": "boolean"
          },
          "default": {},
          "description": {
            "type": "string"
          },
          "enum": {
            "type": "array"
          },
          "items": {
            "properties": {
              "enum": {
                "type": "array"
              },
              "type": {
                "enum": [
                  "array",
```

```

        "boolean",
        "integer",
        "number",
        "object",
        "string",
        null
    ],
    "type": "string"
  },
  "type": "object"
},
"maxItems": {
  "$ref": "#/definitions/positiveInteger"
},
"maxLength": {
  "$ref": "#/definitions/positiveInteger"
},
"maximum": {
  "type": "number"
},
"minItems": {
  "$ref": "#/definitions/positiveIntegerDefault0"
},
"minLength": {
  "$ref": "#/definitions/positiveIntegerDefault0"
},
"minimum": {
  "type": "number"
},
"name": {
  "type": "string"
},
"pattern": {
  "format": "regex",
  "type": "string"
},
"readonly": {
  "type": "boolean"
},
"required": {
  "$ref": "#/definitions/stringArray"
},
"title": {
  "type": "string"
},
"type": {
  "enum": [
    "array",
    "boolean",
    "integer",
    "number",
    "object",
    "string",
    null
  ],
  "type": "string"
},
"uniqueItems": {
  "default": false,

```

```

        "type": "boolean"
      },
      "required": [
        "title",
        "type"
      ],
      "type": "object"
    },
    "type": "object"
  },
  "stringArray": {
    "items": {
      "type": "string"
    },
    "type": "array",
    "uniqueItems": true
  }
},
"name": "namespace",
"properties": {
  "created_at": {
    "description": "Date and time of namespace creation (READ-ONLY)",
    "format": "date-time",
    "type": "string"
  },
  "description": {
    "description": "Provides a user friendly description of the
namespace.",
    "maxLength": 500,
    "type": "string"
  },
  "display_name": {
    "description": "The user friendly name for the namespace. Used by
UI if available.",
    "maxLength": 80,
    "type": "string"
  },
  "namespace": {
    "description": "The unique namespace text.",
    "maxLength": 80,
    "type": "string"
  },
  "objects": {
    "items": {
      "properties": {
        "description": {
          "type": "string"
        },
        "name": {
          "type": "string"
        },
        "properties": {
          "$ref": "#/definitions/property"
        },
        "required": {
          "$ref": "#/definitions/stringArray"
        }
      }
    },
    "type": "object"
  }
}

```

```

        },
        "type": "array"
    },
    "owner": {
        "description": "Owner of the namespace.",
        "maxLength": 255,
        "type": "string"
    },
    "properties": {
        "$ref": "#/definitions/property"
    },
    "protected": {
        "description": "If true, namespace will not be deletable.",
        "type": "boolean"
    },
    "resource_type_associations": {
        "items": {
            "properties": {
                "name": {
                    "type": "string"
                },
                "prefix": {
                    "type": "string"
                },
                "properties_target": {
                    "type": "string"
                }
            }
        },
        "type": "object"
    },
    "type": "array"
},
"schema": {
    "type": "string"
},
"self": {
    "type": "string"
},
"updated_at": {
    "description": "Date and time of the last namespace modification (READ-ONLY)",
    "format": "date-time",
    "type": "string"
},
"visibility": {
    "description": "Scope of namespace accessibility.",
    "enum": [
        "public",
        "private"
    ],
    "type": "string"
}
},
"required": [
    "namespace"
]
}

```

12.12.2. Show metadata definition namespaces schema

Method	URI	Description
GET	/v2/schemas/metadefs/namespaces	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespaces entity.

A namespaces entity is a container for namespace entities.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

12.12.2.1. Request

This operation does not accept a request body.

12.12.2.2. Response

Example 12.45. Show metadata definition namespaces schema: JSON response

```
{
  "definitions": {
    "positiveInteger": {
      "minimum": 0,
      "type": "integer"
    },
    "positiveIntegerDefault0": {
      "allOf": [
        {
          "$ref": "#/definitions/positiveInteger"
        },
        {
          "default": 0
        }
      ]
    }
  },
  "property": {
    "additionalProperties": {
      "properties": {
        "additionalItems": {
          "type": "boolean"
        },
        "default": {},
        "description": {
          "type": "string"
        },
        "enum": {
          "type": "array"
        },
        "items": {
          "properties": {
            "enum": {
              "type": "array"
            }
          },
          "type": {
```



```

        "enum": [
            "array",
            "boolean",
            "integer",
            "number",
            "object",
            "string",
            null
        ],
        "type": "string"
    },
    },
    "type": "object"
},
"maxItems": {
    "$ref": "#/definitions/positiveInteger"
},
"maxLength": {
    "$ref": "#/definitions/positiveInteger"
},
"maximum": {
    "type": "number"
},
"minItems": {
    "$ref": "#/definitions/positiveIntegerDefault0"
},
"minLength": {
    "$ref": "#/definitions/positiveIntegerDefault0"
},
"minimum": {
    "type": "number"
},
"name": {
    "type": "string"
},
"pattern": {
    "format": "regex",
    "type": "string"
},
"readonly": {
    "type": "boolean"
},
"required": {
    "$ref": "#/definitions/stringArray"
},
"title": {
    "type": "string"
},
"type": {
    "enum": [
        "array",
        "boolean",
        "integer",
        "number",
        "object",
        "string",
        null
    ],
    "type": "string"
},
},

```

```

        "uniqueItems": {
            "default": false,
            "type": "boolean"
        },
        "required": [
            "title",
            "type"
        ],
        "type": "object"
    },
    "type": "object"
},
"stringArray": {
    "items": {
        "type": "string"
    },
    "type": "array",
    "uniqueItems": true
},
"links": [
    {
        "href": "{first}",
        "rel": "first"
    },
    {
        "href": "{next}",
        "rel": "next"
    },
    {
        "href": "{schema}",
        "rel": "describedby"
    }
],
"name": "namespaces",
"properties": {
    "first": {
        "type": "string"
    },
    "namespaces": {
        "items": {
            "additionalProperties": false,
            "name": "namespace",
            "properties": {
                "created_at": {
                    "description": "Date and time of namespace creation
(READ-ONLY)",
                    "format": "date-time",
                    "type": "string"
                },
                "description": {
                    "description": "Provides a user friendly description
of the namespace.",
                    "maxLength": 500,
                    "type": "string"
                },
                "display_name": {
                    "description": "The user friendly name for the
namespace. Used by UI if available.",

```

```

        "maxLength": 80,
        "type": "string"
    },
    "namespace": {
        "description": "The unique namespace text.",
        "maxLength": 80,
        "type": "string"
    },
    "objects": {
        "items": {
            "properties": {
                "description": {
                    "type": "string"
                },
                "name": {
                    "type": "string"
                },
                "properties": {
                    "$ref": "#/definitions/property"
                },
                "required": {
                    "$ref": "#/definitions/stringArray"
                }
            },
            "type": "object"
        },
        "type": "array"
    },
    "owner": {
        "description": "Owner of the namespace.",
        "maxLength": 255,
        "type": "string"
    },
    "properties": {
        "$ref": "#/definitions/property"
    },
    "protected": {
        "description": "If true, namespace will not be
deletable.",
        "type": "boolean"
    },
    "resource_type_associations": {
        "items": {
            "properties": {
                "name": {
                    "type": "string"
                },
                "prefix": {
                    "type": "string"
                },
                "properties_target": {
                    "type": "string"
                }
            },
            "type": "object"
        },
        "type": "array"
    },
    "schema": {
        "type": "string"
    }

```

```
    },
    "self": {
      "type": "string"
    },
    "updated_at": {
      "description": "Date and time of the last namespace
modification (READ-ONLY)",
      "format": "date-time",
      "type": "string"
    },
    "visibility": {
      "description": "Scope of namespace accessibility.",
      "enum": [
        "public",
        "private"
      ],
      "type": "string"
    }
  },
  "required": [
    "namespace"
  ],
  "type": "array"
},
"next": {
  "type": "string"
},
"schema": {
  "type": "string"
}
}
```

12.12.3. Show metadata definition object schema

Method	URI	Description
GET	/v2/schemas/metadefs/object	(Since API v2.1) Shows a JSON schema document that represents a metadata definition object entity.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

12.12.3.1. Request

This operation does not accept a request body.

12.12.3.2. Response

Example 12.46. Show metadata definition object schema: JSON response

```
{
  "additionalProperties": false,
  "definitions": {
    "positiveInteger": {
      "minimum": 0,
      "type": "integer"
    },
    "positiveIntegerDefault0": {
      "allOf": [
        {
          "$ref": "#/definitions/positiveInteger"
        },
        {
          "default": 0
        }
      ]
    },
    "property": {
      "additionalProperties": {
        "properties": {
          "additionalItems": {
            "type": "boolean"
          },
          "default": {},
          "description": {
            "type": "string"
          },
          "enum": {
            "type": "array"
          },
          "items": {
            "properties": {
              "enum": {
                "type": "array"
              },
              "type": {
                "enum": [
                  "array",
```

```

        "boolean",
        "integer",
        "number",
        "object",
        "string",
        null
    ],
    "type": "string"
  },
  "type": "object"
},
"maxItems": {
  "$ref": "#/definitions/positiveInteger"
},
"maxLength": {
  "$ref": "#/definitions/positiveInteger"
},
"maximum": {
  "type": "number"
},
"minItems": {
  "$ref": "#/definitions/positiveIntegerDefault0"
},
"minLength": {
  "$ref": "#/definitions/positiveIntegerDefault0"
},
"minimum": {
  "type": "number"
},
"name": {
  "type": "string"
},
"pattern": {
  "format": "regex",
  "type": "string"
},
"readonly": {
  "type": "boolean"
},
"required": {
  "$ref": "#/definitions/stringArray"
},
"title": {
  "type": "string"
},
"type": {
  "enum": [
    "array",
    "boolean",
    "integer",
    "number",
    "object",
    "string",
    null
  ],
  "type": "string"
},
"uniqueItems": {
  "default": false,

```

```

        "type": "boolean"
      }
    },
    "required": [
      "title",
      "type"
    ],
    "type": "object"
  },
  "type": "object"
},
"stringArray": {
  "items": {
    "type": "string"
  },
  "type": "array",
  "uniqueItems": true
}
},
"name": "object",
"properties": {
  "created_at": {
    "description": "Date and time of object creation (READ-ONLY)",
    "format": "date-time",
    "type": "string"
  },
  "description": {
    "type": "string"
  },
  "name": {
    "type": "string"
  },
  "properties": {
    "$ref": "#/definitions/property"
  },
  "required": {
    "$ref": "#/definitions/stringArray"
  },
  "schema": {
    "type": "string"
  },
  "self": {
    "type": "string"
  },
  "updated_at": {
    "description": "Date and time of the last object modification
(READ-ONLY)",
    "format": "date-time",
    "type": "string"
  }
},
"required": [
  "name"
]
}

```

12.12.4. Show metadata definition objects schema

Method	URI	Description
GET	/v2/schemas/metadefs/objects	(Since API v2.1) Shows a JSON schema document that represents a metadata definition objects entity.

An objects entity is a container for object entities.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

12.12.4.1. Request

This operation does not accept a request body.

12.12.4.2. Response

Example 12.47. Show metadata definition objects schema: JSON response

```
{
  "definitions": {
    "positiveInteger": {
      "minimum": 0,
      "type": "integer"
    },
    "positiveIntegerDefault0": {
      "allOf": [
        {
          "$ref": "#/definitions/positiveInteger"
        },
        {
          "default": 0
        }
      ]
    },
    "property": {
      "additionalProperties": {
        "properties": {
          "additionalItems": {
            "type": "boolean"
          },
          "default": {},
          "description": {
            "type": "string"
          },
          "enum": {
            "type": "array"
          },
          "items": {
            "properties": {
              "enum": {
                "type": "array"
              },
              "type": {
                "enum": [
```



```
        "array",
        "boolean",
        "integer",
        "number",
        "object",
        "string",
        null
    ],
    "type": "string"
  },
  "type": "object"
},
"maxItems": {
  "$ref": "#/definitions/positiveInteger"
},
"maxLength": {
  "$ref": "#/definitions/positiveInteger"
},
"maximum": {
  "type": "number"
},
"minItems": {
  "$ref": "#/definitions/positiveIntegerDefault0"
},
"minLength": {
  "$ref": "#/definitions/positiveIntegerDefault0"
},
"minimum": {
  "type": "number"
},
"name": {
  "type": "string"
},
"pattern": {
  "format": "regex",
  "type": "string"
},
"readonly": {
  "type": "boolean"
},
"required": {
  "$ref": "#/definitions/stringArray"
},
"title": {
  "type": "string"
},
"type": {
  "enum": [
    "array",
    "boolean",
    "integer",
    "number",
    "object",
    "string",
    null
  ],
  "type": "string"
},
"uniqueItems": {
```

```

        "default": false,
        "type": "boolean"
    },
    },
    "required": [
        "title",
        "type"
    ],
    "type": "object"
},
"type": "object"
},
"stringArray": {
    "items": {
        "type": "string"
    },
    "type": "array",
    "uniqueItems": true
}
},
"links": [
    {
        "href": "{first}",
        "rel": "first"
    },
    {
        "href": "{next}",
        "rel": "next"
    },
    {
        "href": "{schema}",
        "rel": "describedby"
    }
],
"name": "objects",
"properties": {
    "first": {
        "type": "string"
    },
    "next": {
        "type": "string"
    },
    "objects": {
        "items": {
            "additionalProperties": false,
            "name": "object",
            "properties": {
                "created_at": {
                    "description": "Date and time of object creation
(READ-ONLY)",
                    "format": "date-time",
                    "type": "string"
                },
                "description": {
                    "type": "string"
                },
                "name": {
                    "type": "string"
                }
            },
            "properties": {

```

```
        "$ref": "#/definitions/property"
      },
      "required": {
        "$ref": "#/definitions/stringArray"
      },
      "schema": {
        "type": "string"
      },
      "self": {
        "type": "string"
      },
      "updated_at": {
        "description": "Date and time of the last object
modification (READ-ONLY)",
        "format": "date-time",
        "type": "string"
      }
    },
    "required": [
      "name"
    ],
    "type": "array"
  },
  "schema": {
    "type": "string"
  }
}
```

12.12.5. Show metadata definition property schema

Method	URI	Description
GET	/v2/schemas/metadefs/property	(Since API v2.1) Shows a JSON schema document that represents a metadata definition property entity.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

12.12.5.1. Request

This operation does not accept a request body.

12.12.5.2. Response

Example 12.48. Show metadata definition property schema: JSON response

```
{
  "additionalProperties": false,
  "definitions": {
    "positiveInteger": {
      "minimum": 0,
      "type": "integer"
    },
    "positiveIntegerDefault0": {
      "allOf": [
        {
          "$ref": "#/definitions/positiveInteger"
        },
        {
          "default": 0
        }
      ]
    },
    "stringArray": {
      "items": {
        "type": "string"
      },
      "minItems": 1,
      "type": "array",
      "uniqueItems": true
    }
  },
  "name": "property",
  "properties": {
    "additionalItems": {
      "type": "boolean"
    },
    "default": {},
    "description": {
      "type": "string"
    },
    "enum": {
      "type": "array"
    }
  },
}
```

```
"items": {
  "properties": {
    "enum": {
      "type": "array"
    },
    "type": {
      "enum": [
        "array",
        "boolean",
        "integer",
        "number",
        "object",
        "string",
        null
      ],
      "type": "string"
    }
  },
  "type": "object"
},
"maxItems": {
  "$ref": "#/definitions/positiveInteger"
},
"maxLength": {
  "$ref": "#/definitions/positiveInteger"
},
"maximum": {
  "type": "number"
},
"minItems": {
  "$ref": "#/definitions/positiveIntegerDefault0"
},
"minLength": {
  "$ref": "#/definitions/positiveIntegerDefault0"
},
"minimum": {
  "type": "number"
},
"name": {
  "type": "string"
},
"pattern": {
  "format": "regex",
  "type": "string"
},
"readonly": {
  "type": "boolean"
},
"required": {
  "$ref": "#/definitions/stringArray"
},
"title": {
  "type": "string"
},
"type": {
  "enum": [
    "array",
    "boolean",
    "integer",
    "number",
```

```
        "object",
        "string",
        null
    ],
    "type": "string"
},
"uniqueItems": {
    "default": false,
    "type": "boolean"
}
},
"required": [
    "type",
    "title",
    "name"
]
}
```

12.12.6. Show metadata definition properties schema

Method	URI	Description
GET	/v2/schemas/metadefs/properties	(Since API v2.1) Shows a JSON schema document that represents a metadata definition properties entity.

A properties entity is a container for property entities.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

12.12.6.1. Request

This operation does not accept a request body.

12.12.6.2. Response

Example 12.49. Show metadata definition properties schema: JSON response

```
{
  "definitions": {
    "positiveInteger": {
      "minimum": 0,
      "type": "integer"
    },
    "positiveIntegerDefault0": {
      "allOf": [
        {
          "$ref": "#/definitions/positiveInteger"
        },
        {
          "default": 0
        }
      ]
    },
    "stringArray": {
      "items": {
        "type": "string"
      },
      "minItems": 1,
      "type": "array",
      "uniqueItems": true
    }
  },
  "links": [
    {
      "href": "{first}",
      "rel": "first"
    },
    {
      "href": "{next}",
      "rel": "next"
    },
    {
      "href": "{schema}",

```

```
        "rel": "describedby"
      }
    ],
    "name": "properties",
    "properties": {
      "first": {
        "type": "string"
      },
      "next": {
        "type": "string"
      },
      "properties": {
        "additionalProperties": {
          "additionalProperties": false,
          "name": "property",
          "properties": {
            "additionalItems": {
              "type": "boolean"
            },
            "default": {},
            "description": {
              "type": "string"
            },
            "enum": {
              "type": "array"
            },
            "items": {
              "properties": {
                "enum": {
                  "type": "array"
                },
                "type": {
                  "enum": [
                    "array",
                    "boolean",
                    "integer",
                    "number",
                    "object",
                    "string",
                    null
                  ],
                  "type": "string"
                }
              },
              "type": "object"
            },
            "maxItems": {
              "$ref": "#/definitions/positiveInteger"
            },
            "maxLength": {
              "$ref": "#/definitions/positiveInteger"
            },
            "maximum": {
              "type": "number"
            },
            "minItems": {
              "$ref": "#/definitions/positiveIntegerDefault0"
            },
            "minLength": {
              "$ref": "#/definitions/positiveIntegerDefault0"
            }
          }
        }
      }
    }
  }
}
```



```
    },
    "minimum": {
      "type": "number"
    },
    "name": {
      "type": "string"
    },
    "pattern": {
      "format": "regex",
      "type": "string"
    },
    "readonly": {
      "type": "boolean"
    },
    "required": {
      "$ref": "#/definitions/stringArray"
    },
    "title": {
      "type": "string"
    },
    "type": {
      "enum": [
        "array",
        "boolean",
        "integer",
        "number",
        "object",
        "string",
        null
      ],
      "type": "string"
    },
    "uniqueItems": {
      "default": false,
      "type": "boolean"
    }
  },
  "required": [
    "type",
    "title"
  ],
  "type": "object"
},
"schema": {
  "type": "string"
}
}
```

12.12.7. Show metadata definition tag schema

Method	URI	Description
GET	/v2/schemas/metadefs/tag	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tag entity.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

12.12.7.1. Request

This operation does not accept a request body.

12.12.7.2. Response

Example 12.50. Show metadata definition tag schema: JSON response

```
{
  "additionalProperties": false,
  "required": [
    "name"
  ],
  "name": "tag",
  "properties": {
    "created_at": {
      "type": "string",
      "description": "Date and time of tag creation (READ-ONLY)",
      "format": "date-time"
    },
    "name": {
      "type": "string"
    },
    "updated_at": {
      "type": "string",
      "description": "Date and time of the last tag modification (READ-ONLY)",
      "format": "date-time"
    }
  }
}
```

12.12.8. Show metadata definition tags schema

Method	URI	Description
GET	/v2/schemas/metadefs/tags	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tags entity.

A tags entity is a container for tag entities.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

12.12.8.1. Request

This operation does not accept a request body.

12.12.8.2. Response

Example 12.51. Show metadata definition tags schema: JSON response

```
{
  "name": "tags",
  "links": [
    {
      "href": "{first}",
      "rel": "first"
    },
    {
      "href": "{next}",
      "rel": "next"
    },
    {
      "href": "{schema}",
      "rel": "describedby"
    }
  ],
  "properties": {
    "next": {
      "type": "string"
    },
    "schema": {
      "type": "string"
    },
    "first": {
      "type": "string"
    },
    "tags": {
      "items": {
        "additionalProperties": false,
        "required": [
          "name"
        ],
        "name": "tag",
        "properties": {
          "created_at": {
```

```
        "type": "string",
        "description": "Date and time of tag creation (READ-
ONLY)",
        "format": "date-time"
    },
    "name": {
        "type": "string"
    },
    "updated_at": {
        "type": "string",
        "description": "Date and time of the last tag
modification (READ-ONLY)",
        "format": "date-time"
    }
},
"type": "array"
}
```

12.12.9. Show metadata definition namespace resource type association schema

Method	URI	Description
GET	/v2/schemas/metadefs/resource_type	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type association entity.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

12.12.9.1. Request

This operation does not accept a request body.

12.12.9.2. Response

Example 12.52. Show metadata definition namespace resource type association schema: JSON response

```
{
  "additionalProperties": false,
  "name": "resource_type_association",
  "properties": {
    "created_at": {
      "description": "Date and time of resource type association (READ-ONLY)",
      "format": "date-time",
      "type": "string"
    },
    "name": {
      "description": "Resource type names should be aligned with Heat resource types whenever possible: http://docs.openstack.org/developer/heat/template_guide/openstack.html",
      "maxLength": 80,
      "type": "string"
    },
    "prefix": {
      "description": "Specifies the prefix to use for the given resource type. Any properties in the namespace should be prefixed with this prefix when being applied to the specified resource type. Must include prefix separator (e.g. a colon :). It is important to note that the same base property key can require different prefixes depending on the target resource type. For example: The desired virtual CPU topology can be set on both images and flavors via metadata. The keys have different prefixes on images than on flavors On flavors keys are prefixed with 'hw:', but on images the keys are prefixed with 'hw_'.",
      "maxLength": 80,
      "type": "string"
    },
    "properties_target": {
      "description": "Some resource types allow more than one key / value pair per instance. For example, Cinder allows user and image
```

```
metadata on volumes. Only the image properties metadata is evaluated by Nova
(scheduling or drivers). This property allows a namespace target to remove
the ambiguity.",
    "maxLength": 80,
    "type": "string"
  },
  "updated_at": {
    "description": "Date and time of the last resource type
association modification (READ-ONLY)",
    "format": "date-time",
    "type": "string"
  }
},
"required": [
  "name"
]
}
```

12.12.10. Show metadata definition namespace resource type associations schema

Method	URI	Description
GET	/v2/schemas/metadefs/resource_types	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type associations entity.

A resource type associations entity is a container for resource type association entities.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

12.12.10.1. Request

This operation does not accept a request body.

12.12.10.2. Response

Example 12.53. Show metadata definition namespace resource type associations schema: JSON response

```
{
  "links": [
    {
      "href": "{first}",
      "rel": "first"
    },
    {
      "href": "{next}",
      "rel": "next"
    },
    {
      "href": "{schema}",
      "rel": "describedby"
    }
  ],
  "name": "resource_type_associations",
  "properties": {
    "first": {
      "type": "string"
    },
    "next": {
      "type": "string"
    },
    "resource_type_associations": {
      "items": {
        "additionalProperties": false,
        "name": "resource_type_association",
        "properties": {
          "created_at": {
```

```

        "description": "Date and time of resource type
association (READ-ONLY)",
        "format": "date-time",
        "type": "string"
    },
    "name": {
        "description": "Resource type names should be aligned
with Heat resource types whenever possible: http://docs.openstack.org/
developer/heat/template_guide/openstack.html",
        "maxLength": 80,
        "type": "string"
    },
    "prefix": {
        "description": "Specifies the prefix to use for the
given resource type. Any properties in the namespace should be prefixed with
this prefix when being applied to the specified resource type. Must include
prefix separator (e.g. a colon :). It is important to note that the same base
property key can require different prefixes depending on the target resource
type. For example: The desired virtual CPU topology can be set on both images
and flavors via metadata. The keys have different prefixes on images than on
flavors On flavors keys are prefixed with 'hw:', but on images the keys are
prefixed with 'hw_'.",
        "maxLength": 80,
        "type": "string"
    },
    "properties_target": {
        "description": "Some resource types allow more than
one key / value pair per instance. For example, Cinder allows user and image
metadata on volumes. Only the image properties metadata is evaluated by Nova
(scheduling or drivers). This property allows a namespace target to remove
the ambiguity.",
        "maxLength": 80,
        "type": "string"
    },
    "updated_at": {
        "description": "Date and time of the last resource
type association modification (READ-ONLY)",
        "format": "date-time",
        "type": "string"
    }
},
"required": [
    "name"
]
},
"type": "array"
},
"schema": {
    "type": "string"
}
}
}

```

12.13. Tasks (since API v2.2) (tasks)

Creates, lists, and shows details for tasks.

Method	URI	Description
POST	/v2/tasks	Creates a task.

Method	URI	Description
GET	/v2/tasks{?sort_dir, sort_key, type, status}	Lists tasks.
GET	/v2/tasks/{task_id}	Shows details for a task.

12.13.1. Create task

Method	URI	Description
POST	/v2/tasks	Creates a task.

Normal response codes: 201

12.13.1.1. Request

Example 12.54. Create task: JSON request

```
{
  "type": "import",
  "input": {
    "import_from": "http://example.com",
    "import_from_format": "qcow2",
    "image_properties": {
      "disk_format": "vhd",
      "container_format": "ovf"
    }
  }
}
```

12.13.1.2. Response

Example 12.55. Create task: JSON response

```
{
  "id": "e7e59ff6-fa2e-4075-87d3-1a1398a07dc3",
  "type": "import",
  "status": "pending"
}
```

12.13.2. List tasks

Method	URI	Description
GET	/v2/tasks{?sort_dir, sort_key, type, status}	Lists tasks.

Normal response codes: 200

12.13.2.1. Request

This table shows the query parameters for the list tasks request:

Name	Type	Description
sort_dir	String (Optional)	Sorts the response by a set of one or more sort direction and attribute (sort_key) combinations. A valid value for the sort direction is asc (ascending) or desc (descending). If you omit the sort direction in a set, the default is desc.
sort_key	String (Optional)	Sorts the response by an attribute, such as name, id, or updated_at. Default is created_at. The API uses the natural sorting direction of the sort_key image attribute.
type	String (Optional)	Filters the response by a task type. A valid value is import.
status	String (Optional)	Filters the response by a task status. A valid value is pending, processing, success, or failure.

This operation does not accept a request body.

12.13.2.2. Response

Example 12.56. List tasks: JSON response

```
{
  "tasks": [
    {
      "id": "cbc36478b0bd8e67e89469c7749d4127",
      "type": "import",
      "status": "pending"
    },
    {
      "id": "bbc36578b0bd8e67e89469c7749d4126",
      "type": "import",
      "status": "processing"
    }
  ]
}
```

12.13.3. Show task details

Method	URI	Description
GET	/v2/tasks/{task_id}	Shows details for a task.

Normal response codes: 200

12.13.3.1. Request

This table shows the URI parameters for the show task details request:

Name	Type	Description
{task_id}	UUID	The UUID of the task.

This operation does not accept a request body.

12.13.3.2. Response

Example 12.57. Show task details: JSON response

```
{
  "id": "e7e59ff6-fa2e-4075-87d3-1a1398a07dc3",
  "type": "import",
  "status": "pending"
}
```

13. Image service API v1 (SUPPORTED)

Loads images for use at launch time by the Compute API. Also, assigns metadata to images.

Some cloud implementations do not expose this API and offer pretested images only.

Cloud providers can configure property protections that prevent non-administrative users from updating and deleting protected properties. For more information, see [Image property protection](#) in the *OpenStack Cloud Administrator Guide*.

Method	URI	Description
API versions		
GET	/	Lists information about all Image service API versions.
GET	/versions	Shows details for the Image service API v1.
Images		
GET	/v1/images{?name,container_format,disk_format,status,size_min,size_max,changes-since}	Lists all public VM images.
POST	/v1/images	Registers a virtual machine (VM) image.
GET	/v1/images/detail{?name,container_format,disk_format,status,size_min,size_max,changes-since}	Lists all available images with details.
GET	/v1/images/{image_id}	Shows the image details as headers and the image binary in the body of the response.
HEAD	/v1/images/{image_id}	Shows the image metadata information in the body of the response.
PUT	/v1/images/{image_id}	Updates an image, uploads an image file, or updates metadata for an image.
DELETE	/v1/images/{image_id}	Deletes an image.
Members		
PUT	/v1/images/{image_id}/members	Replaces a membership list for an image.
PUT	/v1/images/{image_id}/members/{owner_id}	Adds one or more members to an image.
DELETE	/v1/images/{image_id}/members/{owner_id}	Removes a member from an image.
Shared images		
GET	/v1/shared-images/{owner_id}	Lists the VM images that an owner shares. The owner ID is the tenant ID.

13.1. API versions

Method	URI	Description
GET	/	Lists information about all Image service API versions.
GET	/versions	Shows details for the Image service API v1.

13.1.1. List API versions

Method	URI	Description
GET	/	Lists information about all Image service API versions.

Normal response codes: 300

13.1.1.1. Request

This operation does not accept a request body.

13.1.2. List API versions

Method	URI	Description
GET	/versions	Shows details for the Image service API v1.

Normal response codes: 200

13.1.2.1. Request

This operation does not accept a request body.

13.1.2.2. Response

Example 13.1. List API versions: JSON response

```
{
  "versions": [
    {
      "status": "EXPERIMENTAL",
      "id": "v3.0",
      "links": [
        {
          "href": "http://23.253.228.211:9292/v3/",
          "rel": "self"
        }
      ]
    },
    {
      "status": "CURRENT",
      "id": "v2.3",
      "links": [
        {
          "href": "http://23.253.228.211:9292/v2/",
          "rel": "self"
        }
      ]
    },
    {
      "status": "SUPPORTED",
      "id": "v2.2",
      "links": [
        {
          "href": "http://23.253.228.211:9292/v2/",
          "rel": "self"
        }
      ]
    },
    {
      "status": "SUPPORTED",
      "id": "v2.1",
      "links": [
        {
          "href": "http://23.253.228.211:9292/v2/",
          "rel": "self"
        }
      ]
    }
  ]
}
```

```

    },
    {
      "status": "SUPPORTED",
      "id": "v2.0",
      "links": [
        {
          "href": "http://23.253.228.211:9292/v2/",
          "rel": "self"
        }
      ]
    }
  ],
  {
    "status": "SUPPORTED",
    "id": "v1.1",
    "links": [
      {
        "href": "http://23.253.228.211:9292/v1/",
        "rel": "self"
      }
    ]
  },
  {
    "status": "SUPPORTED",
    "id": "v1.0",
    "links": [
      {
        "href": "http://23.253.228.211:9292/v1/",
        "rel": "self"
      }
    ]
  }
]
}

```

13.2. Images

Method	URI	Description
GET	/v1/images{?name,container_format,disk_format,status,size_min,size_max,changes-since}	Lists all public VM images.
POST	/v1/images	Registers a virtual machine (VM) image.
GET	/v1/images/detail{?name,container_format,disk_format,status,size_min,size_max,changes-since}	Lists all available images with details.
GET	/v1/images/{image_id}	Shows the image details as headers and the image binary in the body of the response.
HEAD	/v1/images/{image_id}	Shows the image metadata information in the body of the response.
PUT	/v1/images/{image_id}	Updates an image, uploads an image file, or updates metadata for an image.
DELETE	/v1/images/{image_id}	Deletes an image.

13.2.1. List images

Method	URI	Description
GET	/v1/images{?name,container_format,disk_format,status,size_min,size_max,changes-since}	Lists all public VM images.

Normal response codes: 200

13.2.1.1. Request

This table shows the query parameters for the list images request:

Name	Type	Description
name	String (Optional)	Filters the image list by an image name, in string format.
container_format	String (Optional)	Filters the image list by a container format. A valid value is aki, ami, ari, bare, docker, ova, or ovf.
disk_format	String (Optional)	Filters the image list by a disk format. A valid value is aki, ami, ari, iso, qcow2, raw, vhd, vdi, or vmdk.
status	String (Optional)	Filters the image list by a status. A valid value is queued, saving, active, killed, deleted, or pending_delete.
size_min	String (Optional)	Filters the image list by a minimum image size, in bytes.
size_max	String (Optional)	Filters the image list by a maximum image size, in bytes.
changes-since	DateTime (Optional)	Filters the image list to those images that have changed since a time stamp value.

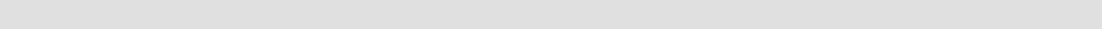
This operation does not accept a request body.

13.2.1.2. Response

Example 13.2. List images: JSON response

```
{
  "images": [
    {
      "uri": "http://glance.example.com/images/71c675ab-d94f-49cd-a114-e12490b328d9",
      "name": "Ubuntu 10.04 Plain",
      "disk_format": "vhd",
      "container_format": "ovf",
      "size": "5368709120"
    },
    {
      "...": "..."
    }
  ]
}
```

```
}
```



13.2.2. Create image

Method	URI	Description
POST	/v1/images	Registers a virtual machine (VM) image.

Normal response codes: 202

13.2.2.1. Request

This operation does not accept a request body.

13.2.2.2. Response

This operation does not return a response body.

13.2.3. List images with details

Method	URI	Description
GET	/v1/images/detail{?name, container_format, disk_format, status, size_min, size_max, changes-since}	Lists all available images with details.

Normal response codes: 200

13.2.3.1. Request

This table shows the query parameters for the list images with details request:

Name	Type	Description
name	String (Optional)	Filters the image list by an image name, in string format.
container_format	String (Optional)	Filters the image list by a container format. A valid value is aki, ami, ari, bare, docker, ova, or ovf.
disk_format	String (Optional)	Filters the image list by a disk format. A valid value is aki, ami, ari, iso, qcow2, raw, vhd, vdi, or vmdk.
status	String (Optional)	Filters the image list by a status. A valid value is queued, saving, active, killed, deleted, or pending_delete.
size_min	String (Optional)	Filters the image list by a minimum image size, in bytes.
size_max	String (Optional)	Filters the image list by a maximum image size, in bytes.
changes-since	DateTime (Optional)	Filters the image list to those images that have changed since a time stamp value.

This operation does not accept a request body.

13.2.3.2. Response

Example 13.3. List images with details: JSON response

```
{
  "images": [
    {
      "uri": "http://glance.example.com/images/71c675ab-d94f-49cd-a114-e12490b328d9",
      "name": "Ubuntu 10.04 Plain 5GB",
      "disk_format": "vhd",
      "container_format": "ovf",
      "size": "5368709120",
      "checksum": "c2e5db72bd7fd153f53ede5da5a06de3",
      "created_at": "2010-02-03 09:34:01",
      "updated_at": "2010-02-03 09:34:01",
      "deleted_at": ""
    }
  ]
}
```

```
        "status": "active",
        "is_public": true,
        "min_ram": 256,
        "min_disk": 5,
        "owner": null,
        "properties": {
            "distro": "Ubuntu 10.04 LTS"
        }
    },
    {
        "...": "..."
    }
]
```

13.2.4. Show image details and image binary

Method	URI	Description
GET	/v1/images/{image_id}	Shows the image details as headers and the image binary in the body of the response.

Normal response codes: 200

Error response codes: forbidden (403)

13.2.4.1. Request

This table shows the URI parameters for the show image details and image binary request:

Name	Type	Description
{image_id}	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

13.2.4.2. Response

Example 13.4. Show image details and image binary: JSON response

```
{
  "images": [
    {
      "uri": "http://glance.example.com/images/71c675ab-d94f-49cd-a114-
e12490b328d9",
      "name": "Ubuntu 10.04 Plain 5GB",
      "disk_format": "vhd",
      "container_format": "ovf",
      "size": "5368709120",
      "checksum": "c2e5db72bd7fd153f53ede5da5a06de3",
      "created_at": "2010-02-03 09:34:01",
      "updated_at": "2010-02-03 09:34:01",
      "deleted_at": "",
      "status": "active",
      "is_public": true,
      "min_ram": 256,
      "min_disk": 5,
      "owner": null,
      "properties": {
        "distro": "Ubuntu 10.04 LTS"
      }
    },
    {
      "...": "..."
    }
  ]
}
```

13.2.5. Show image metadata

Method	URI	Description
HEAD	/v1/images/{image_id}	Shows the image metadata information in the body of the response.

The Image system does not return a response body for the **HEAD** operation.

Example requests and responses:

- Show image metadata:

```
http://glance.example.com/v1/images/03bc0a8b-659c-4de9-b6bd-13c6e86e6455
```

```
X-Image-Meta-Checksum # 8a40c862b5735975d82605c1dd395796
X-Image-Meta-Container_format # aki
X-Image-Meta-Created_at # 2016-01-06T03:22:20.000000
X-Image-Meta-Deleted # false
X-Image-Meta-Disk_format # aki
X-Image-Meta-Id # 03bc0a8b-659c-4de9-b6bd-13c6e86e6455
X-Image-Meta-Is_public # true
X-Image-Meta-Min_disk # 0
X-Image-Meta-Min_ram # 0
X-Image-Meta-Name # cirros-0.3.4-x86_64-uec-kernel
X-Image-Meta-Owner # 13cc6052265b41529e2fd0fc461fa8ef
X-Image-Meta-Protected # false
X-Image-Meta-Size # 4979632
X-Image-Meta-Status # deactivated
X-Image-Meta-Updated_at # 2016-02-25T03:02:05.000000
X-Openstack-Request-Id # req-d5208320-28ed-4c22-b628-12dc6456d983
```

If the request succeeds, the operation returns the 200 response code.

If there is an image size mismatch detected with the X-Image-Meta-Size, the operation returns a 409 response code.

Normal response codes: 200

Error response codes: itemNotFound (404), 409

13.2.5.1. Request

This table shows the URI parameters for the show image metadata request:

Name	Type	Description
{image_id}	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

13.2.6. Update image

Method	URI	Description
PUT	/v1/images/{image_id}	Updates an image, uploads an image file, or updates metadata for an image.

Normal response codes: 200

13.2.6.1. Request

This table shows the URI parameters for the update image request:

Name	Type	Description
{image_id}	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

13.2.6.2. Response

This operation does not return a response body.

13.2.7. Delete image

Method	URI	Description
DELETE	/v1/images/{image_id}	Deletes an image.

Normal response codes: 204

13.2.7.1. Request

This table shows the URI parameters for the delete image request:

Name	Type	Description
{image_id}	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

13.3. Members

Method	URI	Description
PUT	/v1/images/{image_id}/members	Replaces a membership list for an image.
PUT	/v1/images/{image_id}/members/{owner_id}	Adds one or more members to an image.
DELETE	/v1/images/{image_id}/members/{owner_id}	Removes a member from an image.

13.3.1. Replace member

Method	URI	Description
PUT	/v1/images/{image_id}/members	Replaces a membership list for an image.

Normal response codes: 204

13.3.1.1. Request

This table shows the URI parameters for the replace member request:

Name	Type	Description
{image_id}	String	Image ID stored through the image API. Typically a UUID.

13.3.2. Add members to image

Method	URI	Description
PUT	/v1/images/{image_id}/members/{owner_id}	Adds one or more members to an image.

If you omit the request body, this call adds the membership to the image, leaves the existing memberships unmodified, and sets the `can_share` attribute to `false` for new memberships.

Normal response codes: 204

13.3.2.1. Request

This table shows the URI parameters for the add members to image request:

Name	Type	Description
{image_id}	String	Image ID stored through the image API. Typically a UUID.
{owner_id}	String	Owner ID, which is the tenant ID.

Example 13.5. Add members to image: JSON request

```
{
  "members": [
    {
      "member_id": "tenant1",
      "can_share": false
    },
    {
      "member_id": "tenant2",
      "can_share": false
    }
  ]
}
```

13.3.3. Remove member

Method	URI	Description
DELETE	/v1/images/{image_id}/members/{owner_id}	Removes a member from an image.

Normal response codes: 204

13.3.3.1. Request

This table shows the URI parameters for the remove member request:

Name	Type	Description
{image_id}	String	Image ID stored through the image API. Typically a UUID.
{owner_id}	String	Owner ID, which is the tenant ID.

This operation does not accept a request body.

13.4. Shared images

Method	URI	Description
GET	/v1/shared-images/{owner_id}	Lists the VM images that an owner shares. The owner ID is the tenant ID.

13.4.1. List shared images

Method	URI	Description
GET	/v1/shared-images/{owner_id}	Lists the VM images that an owner shares. The owner ID is the tenant ID.

Normal response codes: 200

13.4.1.1. Request

This table shows the URI parameters for the list shared images request:

Name	Type	Description
{owner_id}	String	Owner ID, which is the tenant ID.

This operation does not accept a request body.

13.4.1.2. Response

Example 13.6. List shared images: JSON response

```
{
  "shared_images": [
    {
      "image_id": "71c675ab-d94f-49cd-a114-e12490b328d9",
      "can_share": false
    },
    {
      "...": "..."
    }
  ]
}
```

14. Networking API v2.0 (CURRENT)

Use virtual networking services among devices that the OpenStack Compute service manages. The Networking (neutron) API v2.0 combines the API v1.1 functionality with some essential Internet Protocol Address Management (IPAM) functionality. The API enables users to associate IP address blocks and other network configuration settings with an OpenStack Networking network. You can choose an IP address from the block or let OpenStack Networking choose the first available IP address.

Effective in the OpenStack Liberty release, the Networking API v2.0 does not support XML requests and responses.

Method	URI	Description
API versions		
GET	/	Lists information about all Networking API versions.
Networks		
GET	/v2.0/networks	Lists networks to which the tenant has access.
POST	/v2.0/networks	Creates a network.
POST	/v2.0/networks	Creates multiple networks in a single request.
GET	/v2.0/networks/{network_id}	Shows details for a network.
PUT	/v2.0/networks/{network_id}	Updates a network.
DELETE	/v2.0/networks/{network_id}	Deletes a network and its associated resources.
Subnets		
GET	/v2.0/subnets{?display_name, network_id, gateway_ip, ip_version, cidr, id, enable_dhcp, ipv6_ra_mode, ipv6_address_mode}	Lists subnets to which the tenant has access.
POST	/v2.0/subnets	Creates a subnet on a network.
POST	/v2.0/subnets	Creates multiple subnets in a single request. Specify a list of subnets in the request body.
GET	/v2.0/subnets/{subnet_id}	Shows details for a subnet.
PUT	/v2.0/subnets/{subnet_id}	Updates a subnet.
DELETE	/v2.0/subnets/{subnet_id}	Deletes a subnet.
Ports		
GET	/v2.0/ports{?status, display_name, admin_state, network_id, tenant_id, device_owner, mac_address, port_id, security_groups, device_id}	Lists ports to which the tenant has access.
POST	/v2.0/ports	Creates a port on a network.
POST	/v2.0/ports	Creates multiple ports in a single request. Specify a list of ports in the request body.
GET	/v2.0/ports/{port_id}	Shows details for a port.
PUT	/v2.0/ports/{port_id}	Updates a port.
DELETE	/v2.0/ports/{port_id}	Deletes a port.
Service providers		
GET	/v2.0/service-providers	Lists service providers and their associated service types.

14.1. API versions

Lists information for all Networking API versions.

Method	URI	Description
GET	/	Lists information about all Networking API versions.

14.1.1. List API versions

Method	URI	Description
GET	/	Lists information about all Networking API versions.

Normal response codes: 200300

14.1.1.1. Request

This operation does not accept a request body.

14.1.1.2. Response

Example 14.1. List API versions: JSON response

```
{
  "versions": [
    {
      "status": "CURRENT",
      "id": "v2.0",
      "links": [
        {
          "href": "http://23.253.228.211:9696/v2.0",
          "rel": "self"
        }
      ]
    }
  ]
}
```

14.2. Networks

Lists, shows details for, creates, updates, and deletes networks.

Method	URI	Description
GET	/v2.0/networks	Lists networks to which the tenant has access.
POST	/v2.0/networks	Creates a network.
POST	/v2.0/networks	Creates multiple networks in a single request.
GET	/v2.0/networks/{network_id}	Shows details for a network.
PUT	/v2.0/networks/{network_id}	Updates a network.
DELETE	/v2.0/networks/{network_id}	Deletes a network and its associated resources.

14.2.1. List networks

Method	URI	Description
GET	/v2.0/networks	Lists networks to which the tenant has access.

Use the `fields` query parameter to filter the response. For information, see [Filtering and Column Selection](#).

Use the `tags`, `tags-any`, `not-tags`, `not-tags-any` query parameter to filter the response with tags. For information, see [REST API Impact](#).

Normal response codes: 200

Error response codes: unauthorized (401)

14.2.1.1. Request

This operation does not accept a request body.

14.2.1.2. Response

Example 14.2. List networks: JSON response

```
{
  "networks": [
    {
      "status": "ACTIVE",
      "subnets": [
        "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
      ],
      "name": "private-network",
      "provider:physical_network": null,
      "admin_state_up": true,
      "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
      "provider:network_type": "local",
      "router:external": true,
      "mtu": 0,
      "shared": true,
      "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
      "provider:segmentation_id": null
    },
    {
      "status": "ACTIVE",
      "subnets": [
        "08eae331-0402-425a-923c-34f7cfe39c1b"
      ],
      "name": "private",
      "provider:physical_network": null,
      "admin_state_up": true,
      "tenant_id": "26a7980765d0414dbc1fc1f88cdb7e6e",
      "provider:network_type": "local",
      "router:external": true,
      "mtu": 0,
      "shared": true,
      "id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",

```

```
        "provider:segmentation_id": null
      }
    ]
  }
```

14.2.2. Create network

Method	URI	Description
POST	/v2.0/networks	Creates a network.

A request body is optional. An administrative user can specify another tenant UUID, which is the tenant who owns the network, in the request body.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

14.2.2.1. Request

Example 14.3. Create network: JSON request

```
{
  "network": {
    "name": "sample_network",
    "admin_state_up": true
  }
}
```

14.2.2.2. Response

Example 14.4. Create network: JSON response

```
{
  "network": {
    "status": "ACTIVE",
    "subnets": [],
    "name": "net1",
    "admin_state_up": true,
    "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
    "router:external": false,
    "mtu": 0,
    "shared": false,
    "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c"
  }
}
```

14.2.3. Bulk create networks

Method	URI	Description
POST	/v2.0/networks	Creates multiple networks in a single request.

In the request body, specify a list of networks.

The bulk create operation is always atomic. Either all or no networks in the request body are created.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

14.2.3.1. Request

Example 14.5. Bulk create networks: JSON request

```
{
  "networks": [
    {
      "name": "sample_network3",
      "admin_state_up": true
    },
    {
      "name": "sample_network4",
      "admin_state_up": true
    }
  ]
}
```

14.2.3.2. Response

Example 14.6. Bulk create networks: JSON response

```
{
  "networks": [
    {
      "status": "ACTIVE",
      "subnets": [],
      "name": "sample_network3",
      "provider:physical_network": null,
      "admin_state_up": true,
      "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
      "mtu": 0,
      "shared": false,
      "id": "bc1a76cb-8767-4c3a-bb95-018b822f2130",
      "provider:segmentation_id": null
    },
    {
      "status": "ACTIVE",
      "subnets": [],
      "name": "sample_network4",
      "provider:physical_network": null,
      "admin_state_up": true,

```

```
    "tenant_id": "4fd44f30292945e481c7b8a0c8908869",  
    "mtu": 0,  
    "shared": false,  
    "id": "af374017-c9ae-4a1d-b799-ab73111476e2",  
    "provider:segmentation_id": null  
  }  
]  
}
```

14.2.4. Show network details

Method	URI	Description
GET	/v2.0/networks/{network_id}	Shows details for a network.

You can control which response parameters are returned by using the fields query parameter. For information, see [Filtering and column selection](#).

The response might show extension response parameters. For information, see [Networks multiple provider extension \(networks\)](#).

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

14.2.4.1. Request

This table shows the URI parameters for the show network details request:

Name	Type	Description
{network_id}	UUID	The UUID of the network.

This operation does not accept a request body.

14.2.4.2. Response

Example 14.7. Show network details: JSON response

```
{
  "network": {
    "status": "ACTIVE",
    "subnets": [
      "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
    ],
    "name": "private-network",
    "router:external": false,
    "admin_state_up": true,
    "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
    "created_at": "2016-03-08T20:19:41",
    "mtu": 0,
    "shared": true,
    "port_security_enabled": true,
    "updated_at": "2016-03-08T20:19:41",
    "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22"
  }
}
```

14.2.5. Update network

Method	URI	Description
PUT	/v2.0/networks/{network_id}	Updates a network.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

14.2.5.1. Request

This table shows the URI parameters for the update network request:

Name	Type	Description
{network_id}	UUID	The UUID of the network.

Example 14.8. Update network: JSON request

```
{
  "network": {
    "name": "sample_network_5_updated"
  }
}
```

14.2.5.2. Response

Example 14.9. Update network: JSON response

```
{
  "network": {
    "status": "ACTIVE",
    "subnets": [],
    "name": "sample_network_5_updated",
    "provider:physical_network": null,
    "admin_state_up": true,
    "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
    "provider:network_type": "local",
    "router:external": false,
    "mtu": 0,
    "shared": false,
    "port_security_enabled": true,
    "id": "1f370095-98f6-4079-be64-6d3d4a6adcc6",
    "provider:segmentation_id": null
  }
}
```

14.2.6. Delete network

Method	URI	Description
DELETE	/v2.0/networks/{network_id}	Deletes a network and its associated resources.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

14.2.6.1. Request

This table shows the URI parameters for the delete network request:

Name	Type	Description
{network_id}	UUID	The UUID of the network.

This operation does not accept a request body.

14.3. Subnets

Lists, shows details for, creates, updates, and deletes subnet resources.

Method	URI	Description
GET	/v2.0/subnets{?display_name, network_id, gateway_ip, ip_version, cidr, id, enable_dhcp, ipv6_ra_mode, ipv6_address_mode}	Lists subnets to which the tenant has access.
POST	/v2.0/subnets	Creates a subnet on a network.
POST	/v2.0/subnets	Creates multiple subnets in a single request. Specify a list of subnets in the request body.
GET	/v2.0/subnets/{subnet_id}	Shows details for a subnet.
PUT	/v2.0/subnets/{subnet_id}	Updates a subnet.
DELETE	/v2.0/subnets/{subnet_id}	Deletes a subnet.

14.3.1. List subnets

Method	URI	Description
GET	/v2.0/subnets{?display_name, network_id, gateway_ip, ip_version, cidr, id, enable_dhcp, ipv6_ra_mode, ipv6_address_mode}	Lists subnets to which the tenant has access.

Default policy settings returns exclusively subnets owned by the tenant submitting the request, unless the request is submitted by a user with administrative rights. You can control which attributes are returned by using the fields query parameter. You can filter results by using query string parameters.

Normal response codes: 200

Error response codes: unauthorized (401)

14.3.1.1. Request

This table shows the query parameters for the list subnets request:

Name	Type	Description
display_name	String (Optional)	Name of the network.
network_id	UUID (Optional)	The ID of the attached network.
gateway_ip	String (Optional)	The gateway IP address.
ip_version	Int (Optional)	The IP version, which is 4 or 6.
cidr	String (Optional)	The CIDR.
id	UUID (Optional)	The ID of the subnet.
enable_dhcp	Boolean (Optional)	If true, DHCP is enabled and If false, DHCP is disabled.
ipv6_ra_mode	String (Optional)	A valid value is dhcpv6-stateful, dhcpv6-stateless, or slaac.
ipv6_address_mode	String (Optional)	A valid value is dhcpv6-stateful, dhcpv6-stateless, or slaac.

14.3.1.2. Response

Example 14.10. List subnets: JSON response

```
{
```

```
"subnets": [
  {
    "name": "private-subnet",
    "enable_dhcp": true,
    "network_id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",
    "tenant_id": "26a7980765d0414dbc1fc1f88cdb7e6e",
    "dns_nameservers": [],
    "allocation_pools": [
      {
        "start": "10.0.0.2",
        "end": "10.0.0.254"
      }
    ],
    "host_routes": [],
    "ip_version": 4,
    "gateway_ip": "10.0.0.1",
    "cidr": "10.0.0.0/24",
    "id": "08eae331-0402-425a-923c-34f7cfe39c1b"
  },
  {
    "name": "my_subnet",
    "enable_dhcp": true,
    "network_id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
    "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
    "dns_nameservers": [],
    "allocation_pools": [
      {
        "start": "192.0.0.2",
        "end": "192.255.255.254"
      }
    ],
    "host_routes": [],
    "ip_version": 4,
    "gateway_ip": "192.0.0.1",
    "cidr": "192.0.0.0/8",
    "id": "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
  }
]
```

14.3.2. Create subnet

Method	URI	Description
POST	/v2.0/subnets	Creates a subnet on a network.

OpenStack Networking does not try to derive the correct IP version from the CIDR. If you do not specify the `gateway_ip` attribute, OpenStack Networking allocates an address from the CIDR for the gateway for the subnet.

To specify a subnet without a gateway, set the `gateway_ip` attribute to `null` in the request body. If you do not specify the `allocation_pools` attribute, OpenStack Networking automatically allocates pools for covering all IP addresses in the CIDR, excluding the address reserved for the subnet gateway. Otherwise, you can explicitly specify allocation pools as shown in the following example.

When you specify both the `allocation_pools` and `gateway_ip` attributes, you must ensure that the gateway IP does not overlap with the allocation pools; otherwise, the call returns the `Conflict (409)` response code.

A subnet can have one or more name servers and host routes. Hosts in this subnet use the name servers. Devices with IP addresses from this subnet, not including the local subnet route, use the host routes.

Specify the `ipv6_ra_mode` and `ipv6_address_mode` attributes to create subnets that support IPv6 configurations, such as stateless address autoconfiguration (SLAAC), DHCPv6 stateful, and DHCPv6 stateless configurations.

Normal response codes: 201

Error response codes: `badRequest (400)`, `unauthorized (401)`, `forbidden (403)`, `itemNotFound (404)`, `conflict (409)`

14.3.2.1. Request

Example 14.11. Create subnet: JSON request

```
{
  "subnet": {
    "network_id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
    "ip_version": 4,
    "cidr": "10.0.0.1"
  }
}
```

14.3.2.2. Response

Example 14.12. Create subnet: JSON response

```
{
  "subnet": {
    "name": "",
    "enable_dhcp": true,
    "network_id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
  }
}
```

```
    "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
    "dns_nameservers": [],
    "allocation_pools": [
      {
        "start": "192.168.199.2",
        "end": "192.168.199.254"
      }
    ],
    "host_routes": [],
    "ip_version": 4,
    "gateway_ip": "192.168.199.1",
    "cidr": "192.168.199.0/24",
    "id": "3b80198d-4f7b-4f77-9ef5-774d54e17126"
  }
}
```

14.3.3. Bulk create subnet

Method	URI	Description
POST	/v2.0/subnets	Creates multiple subnets in a single request. Specify a list of subnets in the request body.

The bulk create operation is always atomic. Either all or no subnets in the request body are created.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), conflict (409)

14.3.3.1. Request

Example 14.13. Bulk create subnet: JSON request

```
{
  "subnets": [
    {
      "cidr": "192.168.199.0/24",
      "ip_version": 4,
      "network_id": "e6031bc2-901a-4c66-82da-f4c32ed89406"
    },
    {
      "cidr": "10.56.4.0/22",
      "ip_version": 4,
      "network_id": "64239a54-dcc4-4b39-920b-b37c2144effa"
    }
  ]
}
```

14.3.3.2. Response

Example 14.14. Bulk create subnet: JSON response

```
{
  "subnets": [
    {
      "allocation_pools": [
        {
          "end": "192.168.199.254",
          "start": "192.168.199.2"
        }
      ],
      "cidr": "192.168.199.0/24",
      "dns_nameservers": [],
      "enable_dhcp": true,
      "gateway_ip": "192.168.199.1",
      "host_routes": [],
      "id": "0468a7a7-290d-4127-aedd-6c9449775a24",
      "ip_version": 4,
      "name": "",
      "network_id": "e6031bc2-901a-4c66-82da-f4c32ed89406",
    }
  ]
}
```

```
        "tenant_id": "d19231fc08ec4bc4829b668040d34512"
      },
      {
        "allocation_pools": [
          {
            "end": "10.56.7.254",
            "start": "10.56.4.2"
          }
        ],
        "cidr": "10.56.4.0/22",
        "dns_nameservers": [],
        "enable_dhcp": true,
        "gateway_ip": "10.56.4.1",
        "host_routes": [],
        "id": "b0e7435c-1512-45fb-aa9e-9a7c5932fb30",
        "ip_version": 4,
        "name": "",
        "network_id": "64239a54-dcc4-4b39-920b-b37c2144effa",
        "tenant_id": "d19231fc08ec4bc4829b668040d34512"
      }
    ]
  }
```

14.3.4. Show subnet details

Method	URI	Description
GET	/v2.0/subnets/{subnet_id}	Shows details for a subnet.

Use the fields query parameter to filter the results.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

14.3.4.1. Request

This table shows the URI parameters for the show subnet details request:

Name	Type	Description
{subnet_id}	UUID	The UUID of the subnet.

This operation does not accept a request body.

14.3.4.2. Response

Example 14.15. Show subnet details: JSON response

```
{
  "subnet": {
    "name": "my_subnet",
    "enable_dhcp": true,
    "network_id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
    "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
    "created_at": "2016-03-08T20:19:41",
    "dns_nameservers": [],
    "allocation_pools": [
      {
        "start": "192.0.0.2",
        "end": "192.255.255.254"
      }
    ],
    "host_routes": [],
    "ip_version": 4,
    "gateway_ip": "192.0.0.1",
    "cidr": "192.0.0.0/8",
    "updated_at": "2016-03-08T20:19:41",
    "id": "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
  }
}
```

14.3.5. Update subnet

Method	URI	Description
PUT	/v2.0/subnets/{subnet_id}	Updates a subnet.

Some attributes, such as IP version (`ip_version`), and CIDR (`cidr`) cannot be updated. Attempting to update these attributes results in a 400 Bad Request error.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

14.3.5.1. Request

This table shows the URI parameters for the update subnet request:

Name	Type	Description
{subnet_id}	UUID	The UUID of the subnet.

Example 14.16. Update subnet: JSON request

```
{
  "subnet": {
    "name": "my_subnet"
  }
}
```

14.3.5.2. Response

Example 14.17. Update subnet: JSON response

```
{
  "subnet": {
    "name": "my_subnet",
    "enable_dhcp": true,
    "network_id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",
    "tenant_id": "26a7980765d0414dbc1fc1f88cdb7e6e",
    "dns_nameservers": [],
    "allocation_pools": [
      {
        "start": "10.0.0.2",
        "end": "10.0.0.254"
      }
    ],
    "host_routes": [],
    "ip_version": 4,
    "gateway_ip": "10.0.0.1",
    "cidr": "10.0.0.0/24",
    "id": "08eae331-0402-425a-923c-34f7cfe39c1b"
  }
}
```


14.3.6. Delete subnet

Method	URI	Description
DELETE	/v2.0/subnets/{subnet_id}	Deletes a subnet.

The operation fails if subnet IP addresses are still allocated.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

14.3.6.1. Request

This table shows the URI parameters for the delete subnet request:

Name	Type	Description
{subnet_id}	UUID	The UUID of the subnet.

This operation does not accept a request body.

14.4. Ports

Lists, shows details for, creates, updates, and deletes ports.

Method	URI	Description
GET	/v2.0/ports{?status,display_name,admin_state,network_id,tenant_id,device_owner,mac_address,port_id,security_groups,device_id}	Lists ports to which the tenant has access.
POST	/v2.0/ports	Creates a port on a network.
POST	/v2.0/ports	Creates multiple ports in a single request. Specify a list of ports in the request body.
GET	/v2.0/ports/{port_id}	Shows details for a port.
PUT	/v2.0/ports/{port_id}	Updates a port.
DELETE	/v2.0/ports/{port_id}	Deletes a port.

14.4.1. List ports

Method	URI	Description
GET	/v2.0/ports{?status,display_name,admin_state,network_id,tenant_id,device_owner,mac_address,port_id,security_groups,device_id}	Lists ports to which the tenant has access.

Default policy settings return only those ports that are owned by the tenant who submits the request, unless the request is submitted by a user with administrative rights. Users can control which attributes are returned by using the fields query parameter. You can use query parameters to filter the response. For information, see [Filtering and Column Selection](#).

Normal response codes: 200

Error response codes: unauthorized (401)

14.4.1.1. Request

This table shows the query parameters for the list ports request:

Name	Type	Description
status	String (Optional)	The port status. Value is ACTIVE or DOWN.
display_name	String (Optional)	The port name.
admin_state	Boolean (Optional)	The administrative state of the router, which is up (true) or down (false).
network_id	UUID (Optional)	The UUID of the attached network.
tenant_id	UUID (Optional)	The UUID of the tenant who owns the network. Only administrative users can specify a tenant UUID other than their own. You cannot change this value through authorization policies.
device_owner	String (Optional)	The UUID of the entity that uses this port. For example, a DHCP agent.
mac_address	String (Optional)	The MAC address of the port.
port_id	UUID (Optional)	The UUID of the port.
security_groups	List (Optional)	The UUIDs of any attached security groups.
device_id	UUID (Optional)	The UUID of the device that uses this port. For example, a virtual server.

14.4.1.2. Response

Example 14.18. List ports: JSON response

```
{
  "ports": [
    {
      "status": "ACTIVE",
      "name": "",
      "allowed_address_pairs": [],
      "admin_state_up": true,
      "network_id": "70c1db1f-b701-45bd-96e0-a313ee3430b3",
      "tenant_id": "",
      "extra_dhcp_opts": [],
      "device_owner": "network:router_gateway",
      "mac_address": "fa:16:3e:58:42:ed",
      "fixed_ips": [
        {
          "subnet_id": "008ba151-0b8c-4a67-98b5-0d2b87666062",
          "ip_address": "172.24.4.2"
        }
      ],
      "id": "d80b1a3b-4fc1-49f3-952e-1e2ab7081d8b",
      "security_groups": [],
      "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824"
    },
    {
      "status": "ACTIVE",
      "name": "",
      "allowed_address_pairs": [],
      "admin_state_up": true,
      "network_id": "f27aa545-cbdd-4907-b0c6-c9e8b039dcc2",
      "tenant_id": "d397de8a63f341818f198abb0966f6f3",
      "extra_dhcp_opts": [],
      "device_owner": "network:router_interface",
      "mac_address": "fa:16:3e:bb:3c:e4",
      "fixed_ips": [
        {
          "subnet_id": "288bf4a1-51ba-43b6-9d0a-520e9005db17",
          "ip_address": "10.0.0.1"
        }
      ],
      "id": "f71a6703-d6de-4be1-a91a-a570ede1d159",
      "security_groups": [],
      "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824"
    }
  ]
}
```

14.4.2. Create port

Method	URI	Description
POST	/v2.0/ports	Creates a port on a network.

To define the network in which to create the port, specify the `network_id` attribute in the request body.

Normal response codes: 201

Error response codes: `badRequest` (400), `unauthorized` (401), `forbidden` (403), `itemNotFound` (404), `macGenerationFailure` (503), `serviceUnavailable` (503)

14.4.2.1. Request

Example 14.19. Create port: JSON request

```
{
  "port": {
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "name": "private-port",
    "admin_state_up": true
  }
}
```

14.4.2.2. Response

Example 14.20. Create port: JSON response

```
{
  "port": {
    "status": "DOWN",
    "name": "private-port",
    "allowed_address_pairs": [],
    "admin_state_up": true,
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "tenant_id": "d6700c0c9ffa4f1cb322cd4a1f3906fa",
    "device_owner": "",
    "mac_address": "fa:16:3e:c9:cb:f0",
    "fixed_ips": [
      {
        "subnet_id": "a0304c3a-4f08-4c43-88af-d796509c97d2",
        "ip_address": "10.0.0.2"
      }
    ],
    "id": "65c0ee9f-d634-4522-8954-51021b570b0d",
    "security_groups": [
      "f0ac4394-7e4a-4409-9701-ba8be283dbc3"
    ],
    "device_id": ""
  }
}
```

14.4.3. Bulk create ports

Method	URI	Description
POST	/v2.0/ports	Creates multiple ports in a single request. Specify a list of ports in the request body.

Guarantees the atomic completion of the bulk operation.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), conflict (409), macGenerationFailure (503)

14.4.3.1. Request

Example 14.21. Bulk create ports: JSON request

```
{
  "ports": [
    {
      "name": "sample_port_1",
      "admin_state_up": false,
      "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7"
    },
    {
      "name": "sample_port_2",
      "admin_state_up": false,
      "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7"
    }
  ]
}
```

14.4.3.2. Response

Example 14.22. Bulk create ports: JSON response

```
{
  "ports": [
    {
      "status": "DOWN",
      "name": "sample_port_1",
      "allowed_address_pairs": [],
      "admin_state_up": false,
      "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
      "tenant_id": "d6700c0c9ffa4f1cb322cd4a1f3906fa",
      "device_owner": "",
      "mac_address": "fa:16:3e:48:b8:9f",
      "fixed_ips": [
        {
          "subnet_id": "a0304c3a-4f08-4c43-88af-d796509c97d2",
          "ip_address": "10.0.0.5"
        }
      ],
      "id": "94225baa-9d3f-4b93-bf12-b41e7ce49cdb",
      "security_groups": [

```

```
        "f0ac4394-7e4a-4409-9701-ba8be283dbc3"
      ],
      "device_id": ""
    },
    {
      "status": "DOWN",
      "name": "sample_port_2",
      "allowed_address_pairs": [],
      "admin_state_up": false,
      "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
      "tenant_id": "d6700c0c9ffa4f1cb322cd4a1f3906fa",
      "device_owner": "",
      "mac_address": "fa:16:3e:f4:73:df",
      "fixed_ips": [
        {
          "subnet_id": "a0304c3a-4f08-4c43-88af-d796509c97d2",
          "ip_address": "10.0.0.6"
        }
      ],
      "id": "235b09e0-63c4-47f1-b221-66ba54c21760",
      "security_groups": [
        "f0ac4394-7e4a-4409-9701-ba8be283dbc3"
      ],
      "device_id": ""
    }
  ]
}
```

14.4.4. Show port details

Method	URI	Description
GET	/v2.0/ports/{port_id}	Shows details for a port.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

14.4.4.1. Request

This table shows the URI parameters for the show port details request:

Name	Type	Description
{port_id}	UUID	The UUID of the port.

This operation does not accept a request body.

14.4.4.2. Response

Example 14.23. Show port details: JSON response

```
{
  "port": {
    "status": "ACTIVE",
    "name": "",
    "allowed_address_pairs": [],
    "admin_state_up": true,
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "tenant_id": "7e02058126cc4950b75f9970368ba177",
    "created_at": "2016-03-08T20:19:41",
    "extra_dhcp_opts": [],
    "device_owner": "network:router_interface",
    "mac_address": "fa:16:3e:23:fd:d7",
    "fixed_ips": [
      {
        "subnet_id": "a0304c3a-4f08-4c43-88af-d796509c97d2",
        "ip_address": "10.0.0.1"
      }
    ],
    "id": "46d4bfb9-b26e-41f3-bd2e-e6dcc1ccedb2",
    "updated_at": "2016-03-08T20:19:41",
    "security_groups": [],
    "device_id": "5e3898d7-11be-483e-9732-b2f5eccd2b2e"
  }
}
```

14.4.5. Update port

Method	URI	Description
PUT	/v2.0/ports/{port_id}	Updates a port.

You can update information for a port, such as its symbolic name and associated IPs. When you update IPs for a port, any previously associated IPs are removed, returned to the respective subnet allocation pools, and replaced by the IPs in the request body. Therefore, this operation replaces the `fixed_ip` attribute when you specify it in the request body. If the updated IP addresses are not valid or are already in use, the operation fails and the existing IP addresses are not removed from the port.

When you update security groups for a port and the operation succeeds, any associated security groups are removed and replaced by the security groups in the request body. Therefore, this operation replaces the `security_groups` attribute when you specify it in the request body. If the security groups are not valid, the operation fails and the existing security groups are not removed from the port.

Normal response codes: 200

Error response codes: `badRequest` (400), `unauthorized` (401), `forbidden` (403), `itemNotFound` (404), `conflict` (409)

14.4.5.1. Request

This table shows the URI parameters for the update port request:

Name	Type	Description
{port_id}	UUID	The UUID of the port.

Example 14.24. Update port: JSON request

```
{
  "port": {
    "name": "test-for-port-update",
    "admin_state_up": true,
    "device_owner": "compute:nova",
    "binding:host_id": "test_for_port_update_host"
  }
}
```

14.4.5.2. Response

Example 14.25. Update port: JSON response

```
{
  "port": {
    "status": "DOWN",
    "binding:host_id": "test_for_port_update_host",
    "allowed_address_pairs": [],
    "extra_dhcp_opts": [],
    "device_owner": "compute:nova",
    "binding:profile": {},
  }
}
```



```
    "fixed_ips": [  
      {  
        "subnet_id": "898dec4a-74df-4193-985f-c76721bcc746",  
        "ip_address": "20.20.0.4"  
      }  
    ],  
    "id": "43c831e0-19ce-4a76-9a49-57b57e69428b",  
    "security_groups": [  
      "ce0179d6-8a94-4f7c-91c2-f3038e2acbd0"  
    ],  
    "device_id": "",  
    "name": "test-for-port-update",  
    "admin_state_up": true,  
    "network_id": "883fc383-5ea1-4c8b-8916-e1ddb0a9f365",  
    "tenant_id": "522eda8d23124b25bf03fe44f1986b74",  
    "binding:vif_details": {},  
    "binding:vnic_type": "normal",  
    "binding:vif_type": "binding_failed",  
    "mac_address": "fa:16:3e:11:11:5e"  
  }  
}
```

14.4.6. Delete port

Method	URI	Description
DELETE	/v2.0/ports/{port_id}	Deletes a port.

Any IP addresses that are associated with the port are returned to the respective subnets allocation pools.

Normal response codes: 204

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

14.4.6.1. Request

This table shows the URI parameters for the delete port request:

Name	Type	Description
{port_id}	UUID	The UUID of the port.

This operation does not accept a request body.

14.5. Service providers

Lists service providers.

Method	URI	Description
GET	/v2.0/service-providers	Lists service providers and their associated service types.

14.5.1. List service providers

Method	URI	Description
GET	/v2.0/service-providers	Lists service providers and their associated service types.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), conflict (409)

14.5.1.1. Request

This operation does not accept a request body.

14.5.1.2. Response

Example 14.26. List service providers: JSON response

```
{
  "service_providers": [
    {
      "service_type": "LOADBALANCER",
      "default": true,
      "name": "haproxy"
    }
  ]
}
```

15. Networking API v2.0 extensions (CURRENT)

Method	URI	Description
Extensions		
GET	/v2.0/extensions	Lists available extensions.
GET	/v2.0/extensions/{alias}	Shows details for an extension, by alias.
Networking Flavors Framework v2.0 (CURRENT) (flavor, service_profile)		
GET	/v2.0/flavors	Lists all flavors visible for the tenant account.
POST	/v2.0/flavors	Creates a flavor.
GET	/v2.0/flavors/{flavor_id}	Shows details for a flavor.
PUT	/v2.0/flavors/{flavor_id}	Updates a flavor.
DELETE	/v2.0/flavors/{flavor_id}	Deletes a flavor.
POST	/v2.0/flavors/{flavor_id}/service_profiles	Associate a flavor with a service profile.
DELETE	/v2.0/flavors/{flavor_id}/service_profiles/{profile_id}	Disassociate a flavor from a service profile.
GET	/v2.0/service_profiles	Lists all service profiles visible for the tenant account.
POST	/v2.0/service_profiles	Creates a service profile.
GET	/v2.0/service_profiles/{profile_id}	Shows details for a service profile.
PUT	/v2.0/service_profiles/{profile_id}	Updates a service profile.
DELETE	/v2.0/service_profiles/{profile_id}	Deletes a service profile.
FWaaS v2.0 (CURRENT) (fw, firewalls, firewall_policies, firewall_rules)		
GET	/v2.0/fw/firewalls	Lists all firewalls.
POST	/v2.0/fw/firewalls	Creates a firewall.
GET	/v2.0/fw/firewalls/{firewall_id}	Shows details for a firewall.
PUT	/v2.0/fw/firewalls/{firewall_id}	Updates a firewall.
DELETE	/v2.0/fw/firewalls/{firewall_id}	Deletes a firewall.
GET	/v2.0/fw/firewall_policies	Lists all firewall policies.
POST	/v2.0/fw/firewall_policies	Creates a firewall policy.
GET	/v2.0/fw/firewall_policies/{firewall_policy_id}	Shows details for a firewall policy.
PUT	/v2.0/fw/firewall_policies/{firewall_policy_id}	Updates a firewall policy.
DELETE	/v2.0/fw/firewall_policies/{firewall_policy_id}	Deletes a firewall policy.
PUT	/v2.0/fw/firewall_policies/{firewall_policy_id}/insert_rule	Insert firewall rule into a policy.
PUT	/v2.0/fw/firewall_policies/{firewall_policy_id}/remove_rule	Remove firewall rule from a policy.
GET	/v2.0/fw/firewall_rules	Lists all firewall rules.
POST	/v2.0/fw/firewall_rules	Creates a firewall rule.
GET	/v2.0/fw/firewall_rules/{firewall_rule_id}	Shows details for a firewall rule.

Method	URI	Description
PUT	/v2.0/fw/firewall_rules/{firewall_rule_id}	Updates a firewall rule.
DELETE	/v2.0/fw/firewall_rules/{firewall_rule_id}	Deletes a firewall rule.
LBaaS 2.0 (STABLE) (lbaas, loadbalancers, listeners, health_monitors, pools, members)		
GET	/v2.0/lbaas/loadbalancers	Lists all load balancers for the tenant account.
POST	/v2.0/lbaas/loadbalancers	Creates a load balancer.
GET	/v2.0/lbaas/loadbalancers/{loadbalancer_id}	Shows details for a load balancer.
PUT	/v2.0/lbaas/loadbalancers/{loadbalancer_id}	Updates a load balancer.
DELETE	/v2.0/lbaas/loadbalancers/{loadbalancer_id}	Removes a load balancer and its associated configuration from the tenant account.
GET	/v2.0/lbaas/loadbalancers/{loadbalancer_id}/statuses	Shows the status tree for a load balancer.
GET	/v2.0/lbaas/listeners	Lists all listeners.
POST	/v2.0/lbaas/listeners	Creates a listener.
GET	/v2.0/lbaas/listeners/{listener_id}	Shows details for a listener.
PUT	/v2.0/lbaas/listeners/{listener_id}	Updates a listener.
DELETE	/v2.0/lbaas/listeners/{listener_id}	Removes a listener.
GET	/v2.0/lbaas/pools	Lists all pools that are associated with your tenant account.
POST	/v2.0/lbaas/pools	Creates a pool.
GET	/v2.0/lbaas/pools/{pool_id}	Shows details for a pool.
PUT	/v2.0/lbaas/pools/{pool_id}	Updates a pool.
DELETE	/v2.0/lbaas/pools/{pool_id}	Removes a pool.
GET	/v2.0/lbaas/pools/{pool_id}/members	Lists members of a pool.
POST	/v2.0/lbaas/pools/{pool_id}/members	Adds a member to a pool.
GET	/v2.0/lbaas/pools/{pool_id}/members/{member_id}	Shows details for a pool member.
PUT	/v2.0/lbaas/pools/{pool_id}/members/{member_id}	Updates attributes for a pool member.
DELETE	/v2.0/lbaas/pools/{pool_id}/members/{member_id}	Removes a member from a pool and its associated configuration from the tenant account.
POST	/v2.0/lbaas/health_monitors	Creates a health monitor.
GET	/v2.0/lbaas/health_monitors	Lists health monitors.
GET	/v2.0/lbaas/health_monitors/{health_monitor_id}	Shows details for a health monitor.
PUT	/v2.0/lbaas/health_monitors/{health_monitor_id}	Updates a health monitor.
DELETE	/v2.0/lbaas/health_monitors/{health_monitor_id}	Removes a health monitor and its associated configuration from the tenant account.
LBaaS 1.0 (DEPRECATED) (lb, vips, health_monitors, pools, members)		
GET	/v2.0/lb/vips	Lists VIPs.
POST	/v2.0/lb/vips	Creates a load balancer VIP.

Method	URI	Description
GET	/v2.0/lb/vips/{vip_id}	Shows details for a VIP.
PUT	/v2.0/lb/vips/{vip_id}	Updates a load balancer VIP.
DELETE	/v2.0/lb/vips/{vip_id}	Deletes a load balancer VIP.
GET	/v2.0/lb/health_monitors	Lists health monitors.
POST	/v2.0/lb/health_monitors	Creates a load balancer health monitor.
GET	/v2.0/lb/health_monitors/{health_monitor_id}	Shows details for a health monitor.
PUT	/v2.0/lb/health_monitors/{health_monitor_id}	Updates a load balancer health monitor.
DELETE	/v2.0/lb/health_monitors/{health_monitor_id}	Deletes a load balancer health monitor.
GET	/v2.0/lb/pools	Lists pools.
POST	/v2.0/lb/pools	Creates a load balancer pool.
GET	/v2.0/lb/pools/{pool_id}	Shows details for a pool.
PUT	/v2.0/lb/pools/{pool_id}	Updates a load balancer pool.
DELETE	/v2.0/lb/pools/{pool_id}	Deletes a load balancer pool.
POST	/v2.0/lb/pools/{pool_id}/health_monitors	Associates a health monitor with a pool.
DELETE	/v2.0/lb/pools/{pool_id}/health_monitors/{health_monitor_id}	Disassociates a health monitor from a pool.
GET	/v2.0/lb/members	Lists members.
POST	/v2.0/lb/members	Creates a load balancer member.
GET	/v2.0/lb/members/{member_id}	Shows details for a member.
PUT	/v2.0/lb/members/{member_id}	Updates a load balancer member.
DELETE	/v2.0/lb/members/{member_id}	Deletes a load balancer member.
Metering labels and rules (metering-labels, metering-label-rules)		
GET	/v2.0/metering/metering-labels	Lists all L3 metering labels that belong to the tenant.
POST	/v2.0/metering/metering-labels	Creates an L3 metering label.
GET	/v2.0/metering/metering-labels/{metering-label-id}	Shows details for a metering label.
DELETE	/v2.0/metering/metering-labels/{metering-label-id}	Deletes an L3 metering label.
GET	/v2.0/metering/metering-label-rules	Lists a summary of all L3 metering label rules that belong to the tenant.
POST	/v2.0/metering/metering-label-rules	Creates an L3 metering label rule.
GET	/v2.0/metering/metering-label-rules/{metering-label-rule-id}	Shows details for a metering label rule.
DELETE	/v2.0/metering/metering-label-rules/{metering-label-rule-id}	Deletes an L3 metering label rule.
Networks provider extended attributes (networks)		
GET	/v2.0/networks	Lists networks that are accessible to the tenant who submits the request.
POST	/v2.0/networks	Creates a network.
GET	/v2.0/networks/{network_id}	Shows details for a network.
PUT	/v2.0/networks/{network_id}	Updates a network.
DELETE	/v2.0/networks/{network_id}	Deletes a network.
Networks multiple provider extension (networks)		

Method	URI	Description
GET	/v2.0/networks	Lists networks that are accessible to the tenant who submits the request. Networks with multiple segments include the segments list in the response.
POST	/v2.0/networks	Creates a network with multiple segment mappings.
GET	/v2.0/networks/{network_id}	Shows details for a network with multiple segments.
VLAN transparency extension (networks)		
GET	/v2.0/networks	Lists networks. The response shows the VLAN transparency attribute.
POST	/v2.0/networks	Creates a VLAN-transparent network.
GET	/v2.0/networks/{network_id}	Shows details for a VLAN-transparent network.
Ports binding extended attributes (ports)		
GET	/v2.0/ports{?status, display_name, admin_state, network_id, tenant_id, device_owner, mac_address, port_id, security_groups, device_id}	Lists ports to which the tenant has access.
POST	/v2.0/ports	Creates a port on a network.
GET	/v2.0/ports/{port_id}	Shows details for a port.
PUT	/v2.0/ports/{port_id}	Updates a port.
DELETE	/v2.0/ports/{port_id}	Deletes a port.
QoS policies (qos)		
GET	/v2.0/qos/policies	Lists all QoS policies that are associated with your tenant account.
POST	/v2.0/qos/policies	Creates a QoS policy.
GET	/v2.0/qos/policies/{policy_id}	Shows details for a QoS policy.
PUT	/v2.0/qos/policies/{policy_id}	Updates a QoS policy.
DELETE	/v2.0/qos/policies/{policy_id}	Deletes a QoS policy.
GET	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules	Lists all bandwidth limit rules for a QoS policy.
POST	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules	Creates a bandwidth limit rule for a QoS policy.
GET	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules/{rule_id}	Shows details for a bandwidth limit rule for a QoS policy.
PUT	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules/{rule_id}	Updates a bandwidth limit rule for a QoS policy.
DELETE	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules/{rule_id}	Deletes a bandwidth limit rule for a QoS policy.
Quotas extension (quotas)		
GET	/v2.0/quotas	Lists quotas for tenants who have non-default quota values.
GET	/v2.0/quotas/{tenant_id}	Lists quotas for a tenant.
PUT	/v2.0/quotas/{tenant_id}	Updates quotas for a tenant. Use when non-default quotas are desired.
DELETE	/v2.0/quotas/{tenant_id}	Resets quotas to default values for a tenant.
Extra routes (routers)		
PUT	/v2.0/routers/{router_id}	Updates extra routes on a router.
Layer-3 networking (routers, floatingips)		
GET	/v2.0/routers	Lists logical routers that the tenant who submits the request can access.
POST	/v2.0/routers	Creates a logical router.

Method	URI	Description
GET	/v2.0/routers/{router_id}	Shows details for a router.
PUT	/v2.0/routers/{router_id}	Updates a logical router.
DELETE	/v2.0/routers/{router_id}	Deletes a logical router and, if present, its external gateway interface.
PUT	/v2.0/routers/{router_id}/add_router_interface	Adds an internal interface to a logical router.
PUT	/v2.0/routers/{router_id}/remove_router_interface	Deletes an internal interface from a logical router.
GET	/v2.0/floatingips	Lists floating IPs that are accessible to the tenant who submits the request.
POST	/v2.0/floatingips	Creates a floating IP, and, if you specify port information, associates the floating IP with an internal port.
GET	/v2.0/floatingips/{floatingip_id}	Shows details for a floating IP.
PUT	/v2.0/floatingips/{floatingip_id}	Updates a floating IP and its association with an internal port.
DELETE	/v2.0/floatingips/{floatingip_id}	Deletes a floating IP and, if present, its associated port.
Security groups (security-groups)		
GET	/v2.0/security-groups	Lists OpenStack Networking security groups to which the tenant has access.
POST	/v2.0/security-groups	Creates an OpenStack Networking security group.
GET	/v2.0/security-groups/{security_group_id}{?verbose, fields}	Shows details for a security group.
PUT	/v2.0/security-groups/{security_group_id}	Updates a security group.
DELETE	/v2.0/security-groups/{security_group_id}	Deletes an OpenStack Networking security group.
Security group rules (security-group-rules)		
GET	/v2.0/security-group-rules	Lists a summary of all OpenStack Networking security group rules that the tenant can access.
POST	/v2.0/security-group-rules	Creates an OpenStack Networking security group rule.
GET	/v2.0/security-group-rules/{security-group-rules-id}	Shows detailed information for a security group rule.
DELETE	/v2.0/security-group-rules/{security-group-rules-id}	Deletes a rule from an OpenStack Networking security group.
Subnet pools extension (subnetpools)		
GET	/v2.0/subnetpools	Lists subnet pools to which the tenant has access.
POST	/v2.0/subnetpools	Creates a subnet pool.
GET	/v2.0/subnetpools/{subnetpool_id}	Shows information for a subnet pool.
PUT	/v2.0/subnetpools/{subnetpool_id}	Updates a subnet pool.
DELETE	/v2.0/subnetpools/{subnetpool_id}	Deletes a subnet pool.
Tag extension (tags)		
PUT	/v2.0/{resource_type}/{resource_id}/tags	Replaces all tags on the resource.
DELETE	/v2.0/{resource_type}/{resource_id}/tags	Removes all tags on the resource.
GET	/v2.0/{resource_type}/{resource_id}/tags/{tag}	Confirms a given tag is set on the resource.
PUT	/v2.0/{resource_type}/{resource_id}/tags/{tag}	Adds a tag on the resource.

Method	URI	Description
DELETE	/v2.0/{resource_type}/{resource_id}/tags/{tag}	Removes a tag on the resource.
VPNaaS 2.0 (CURRENT) (vpn, vpnservices, ikepolicies, ipsecpolicies, endpoint-groups, ipsec-site-connections)		
GET	/v2.0/vpn/vpnservices	Lists all VPN services.
POST	/v2.0/vpn/vpnservices	Creates a VPN service.
GET	/v2.0/vpn/vpnservices/{service_id}	Shows details for a VPN service.
PUT	/v2.0/vpn/vpnservices/{service_id}	Updates a VPN service.
DELETE	/v2.0/vpn/vpnservices/{service_id}	Removes a VPN service.
GET	/v2.0/vpn/ikepolicies	Lists IKE policies.
POST	/v2.0/vpn/ikepolicies	Creates an IKE policy.
GET	/v2.0/vpn/ikepolicies/{ikepolicy_id}	Shows details for an IKE policy.
PUT	/v2.0/vpn/ikepolicies/{ikepolicy_id}	Updates policy settings in an IKE policy.
DELETE	/v2.0/vpn/ikepolicies/{ikepolicy_id}	Removes an IKE policy.
GET	/v2.0/vpn/ipsecpolicies	Lists all IPSec policies.
POST	/v2.0/vpn/ipsecpolicies	Creates an IP security (IPSec) policy.
GET	/v2.0/vpn/ipsecpolicies/{ipsecpolicy_id}	Shows details for an IPSec policy.
PUT	/v2.0/vpn/ipsecpolicies/{ipsecpolicy_id}	Updates policy settings in an IPSec policy.
DELETE	/v2.0/vpn/ipsecpolicies/{ipsecpolicy_id}	Removes an IPSec policy.
GET	/v2.0/vpn/endpoint-groups	Lists VPN endpoint groups.
POST	/v2.0/vpn/endpoint-groups	Creates a VPN endpoint group.
GET	/v2.0/vpn/endpoint-groups/{endpoint_group_id}	Shows details for a VPN endpoint group.
PUT	/v2.0/vpn/endpoint-groups/{endpoint_group_id}	Updates settings for a VPN endpoint group.
DELETE	/v2.0/vpn/endpoint-groups/{endpoint_group_id}	Removes a VPN endpoint group.
GET	/v2.0/vpn/ipsec-site-connections	Lists all IPSec connections.
POST	/v2.0/vpn/ipsec-site-connections	Creates a site-to-site IPSec connection for a service.
GET	/v2.0/vpn/ipsec-site-connections/{connection_id}	Shows details for an IPSec connection.
PUT	/v2.0/vpn/ipsec-site-connections/{connection_id}	Updates connection settings for an IPSec connection.
DELETE	/v2.0/vpn/ipsec-site-connections/{connection_id}	Removes an IPSec connection.
Network IP availability and usage stats		
GET	/v2.0/network-ip-availabilities	List network IP availability of all networks.
GET	/v2.0/network-ip-availabilities	Show network IP availability of all networks.

15.1. Extensions

Lists available Networking API v2.0 extensions and shows details for an extension.

Method	URI	Description
GET	/v2.0/extensions	Lists available extensions.
GET	/v2.0/extensions/{alias}	Shows details for an extension, by alias.

15.1.1. List extensions

Method	URI	Description
GET	/v2.0/extensions	Lists available extensions.

Extensions introduce features and vendor-specific functionality to the API.

The response shows the extension name and its alias. To show details for an extension, you specify the alias.

Normal response codes: 200203

15.1.1.1. Request

This operation does not accept a request body.

15.1.1.2. Response

Example 15.1. List extensions: JSON response

```
{
  "extensions": [
    {
      "updated": "2013-01-20T00:00:00-00:00",
      "name": "Neutron Service Type Management",
      "links": [],
      "alias": "service-type",
      "description": "API for retrieving service providers for Neutron
advanced services"
    },
    {
      "updated": "2012-10-05T10:00:00-00:00",
      "name": "security-group",
      "links": [],
      "alias": "security-group",
      "description": "The security groups extension."
    },
    {
      "updated": "2013-02-07T10:00:00-00:00",
      "name": "L3 Agent Scheduler",
      "links": [],
      "alias": "l3_agent_scheduler",
      "description": "Schedule routers among l3 agents"
    },
    {
      "updated": "2013-02-07T10:00:00-00:00",
      "name": "Loadbalancer Agent Scheduler",
      "links": [],
      "alias": "lbaas_agent_scheduler",
      "description": "Schedule pools among lbaas agents"
    },
    {
      "updated": "2013-03-28T10:00:00-00:00",
      "name": "Neutron L3 Configurable external gateway mode",
      "links": [],
      "alias": "ext-gw-mode",

```

```

        "description": "Extension of the router abstraction for specifying
whether SNAT should occur on the external gateway"
    },
    {
        "updated": "2014-02-03T10:00:00-00:00",
        "name": "Port Binding",
        "links": [],
        "alias": "binding",
        "description": "Expose port bindings of a virtual port to external
application"
    },
    {
        "updated": "2012-09-07T10:00:00-00:00",
        "name": "Provider Network",
        "links": [],
        "alias": "provider",
        "description": "Expose mapping of virtual networks to physical
networks"
    },
    {
        "updated": "2013-02-03T10:00:00-00:00",
        "name": "agent",
        "links": [],
        "alias": "agent",
        "description": "The agent management extension."
    },
    {
        "updated": "2012-07-29T10:00:00-00:00",
        "name": "Quota management support",
        "links": [],
        "alias": "quotas",
        "description": "Expose functions for quotas management per tenant"
    },
    {
        "updated": "2013-02-07T10:00:00-00:00",
        "name": "DHCP Agent Scheduler",
        "links": [],
        "alias": "dhcp_agent_scheduler",
        "description": "Schedule networks among dhcp agents"
    },
    {
        "updated": "2013-06-27T10:00:00-00:00",
        "name": "Multi Provider Network",
        "links": [],
        "alias": "multi-provider",
        "description": "Expose mapping of virtual networks to multiple
physical networks"
    },
    {
        "updated": "2013-01-14T10:00:00-00:00",
        "name": "Neutron external network",
        "links": [],
        "alias": "external-net",
        "description": "Adds external network attribute to network
resource."
    },
    {
        "updated": "2012-07-20T10:00:00-00:00",
        "name": "Neutron L3 Router",
        "links": [],

```

```

        "alias": "router",
        "description": "Router abstraction for basic L3 forwarding between
L2 Neutron networks and access to external networks via a NAT gateway."
    },
    {
        "updated": "2013-07-23T10:00:00-00:00",
        "name": "Allowed Address Pairs",
        "links": [],
        "alias": "allowed-address-pairs",
        "description": "Provides allowed address pairs"
    },
    {
        "updated": "2013-03-17T12:00:00-00:00",
        "name": "Neutron Extra DHCP opts",
        "links": [],
        "alias": "extra_dhcp_opt",
        "description": "Extra options configuration for DHCP. For example
PXE boot options to DHCP clients can be specified (e.g. tftp-server, server-
ip-address, bootfile-name)"
    },
    {
        "updated": "2012-10-07T10:00:00-00:00",
        "name": "LoadBalancing service",
        "links": [],
        "alias": "lbaas",
        "description": "Extension for LoadBalancing service"
    },
    {
        "updated": "2013-02-01T10:00:00-00:00",
        "name": "Neutron Extra Route",
        "links": [],
        "alias": "extraroute",
        "description": "Extra routes configuration for L3 router"
    }
]
}

```

15.1.2. Show extension details

Method	URI	Description
GET	/v2.0/extensions/{alias}	Shows details for an extension, by alias.

Normal response codes: 200203

15.1.2.1. Request

This table shows the URI parameters for the show extension details request:

Name	Type	Description
{alias}	String	The alias of an extension.

This operation does not accept a request body.

15.1.2.2. Response

Example 15.2. Show extension details: JSON response

```
{
  "extension": {
    "updated": "2013-02-03T10:00:00-00:00",
    "name": "agent",
    "links": [],
    "alias": "agent",
    "description": "The agent management extension."
  }
}
```

15.2. Networking Flavors Framework v2.0 (CURRENT) (flavor, service_profile)

Extension that allows user selection of operator-curated flavors during resource creation.

Service q-flavors must be enabled in the configuration to use this feature.

Method	URI	Description
GET	/v2.0/flavors	Lists all flavors visible for the tenant account.
POST	/v2.0/flavors	Creates a flavor.
GET	/v2.0/flavors/{flavor_id}	Shows details for a flavor.
PUT	/v2.0/flavors/{flavor_id}	Updates a flavor.
DELETE	/v2.0/flavors/{flavor_id}	Deletes a flavor.
POST	/v2.0/flavors/{flavor_id}/service_profiles	Associate a flavor with a service profile.
DELETE	/v2.0/flavors/{flavor_id}/service_profiles/{profile_id}	Disassociate a flavor from a service profile.
GET	/v2.0/service_profiles	Lists all service profiles visible for the tenant account.
POST	/v2.0/service_profiles	Creates a service profile.

Method	URI	Description
GET	/v2.0/service_profiles/ {profile_id}	Shows details for a service profile.
PUT	/v2.0/service_profiles/ {profile_id}	Updates a service profile.
DELETE	/v2.0/service_profiles/ {profile_id}	Deletes a service profile.

15.2.1. List flavors

Method	URI	Description
GET	/v2.0/flavors	Lists all flavors visible for the tenant account.

The list can be empty.

Standard query parameters are supported on the URI. For example, `fields` can be used to limit the returned response to just name by appending `?fields=name`. If Neutron configuration supports pagination by overriding `allow_pagination = false`, the `marker` query parameter can set the last element id the client has seen and `limit` set the maximum number of items to return. If Neutron configuration has `allow_sorting = true`, `sort_key` and `sort_dir` pairs can be used where sort direction is 'asc' or 'desc'.

Normal response codes: 200

Error response codes: unauthorized (401)

15.2.1.1. Request

This operation does not accept a request body.

15.2.1.2. Response

Example 15.3. List flavors: JSON response

```
{
  "flavors": [
    {
      "description": "",
      "enabled": true,
      "service_profiles": [],
      "service_type": "LOADBALANCERV2",
      "id": "f7b14d9a-b0dc-4fbe-bb14-a0f4970a69e0",
      "name": "dummy"
    }
  ]
}
```


15.2.2. Create flavor

Method	URI	Description
POST	/v2.0/flavors	Creates a flavor.

This operation establishes a new flavor.

The `service_type` to which the flavor applies is a required parameter. The corresponding service plugin must have been activated as part of the configuration. See [Service providers](#) for how to see currently loaded service types. Additionally the service plugin needs to support the use of flavors. For example, the `LOADBALANCERV2` service type using the `LBaaSv2` API currently supports Neutron service flavors.

Creation currently limited to administrators. Other users will receive a `Forbidden 403` response code with a response body `NeutronError` message expressing that creation is disallowed by policy.

Until one or more service profiles are associated with the flavor by the operator, attempts to use the flavor during resource creations will currently return a `Not Found 404` with a response body that indicates no service profile could be found.

If the API cannot fulfill the request due to insufficient data or data that is not valid, the service returns the `HTTP Bad Request (400)` response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.

Normal response codes: 201

Error response codes: `badRequest (400)`, `unauthorized (401)`, `forbidden (403)`

15.2.2.1. Request

Example 15.4. Create flavor: JSON request

```
{
  "flavor": {
    "service_type": "LOADBALANCERV2",
    "enabled": true,
    "name": "dummy",
    "description": "Dummy flavor"
  }
}
```

15.2.2.2. Response

Example 15.5. Create flavor: JSON response

```
{
  "flavor": {
    "id": "7fc0581b-4509-49e1-90eb-c953c877fa4c",
    "name": "dummy",
    "service_type": "LOADBALANCERV2",
    "description": "Dummy flavor",
  }
}
```

```
        "enabled": true,  
        "service_profiles": []  
    }  
}
```

15.2.3. Show flavor details

Method	URI	Description
GET	/v2.0/flavors/{flavor_id}	Shows details for a flavor.

This operation returns a flavor object by ID. If you are not an administrative user and the flavor object is not visible to your tenant account, the service returns the HTTP Forbidden (403) response code.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.2.3.1. Request

This table shows the URI parameters for the show flavor details request:

Name	Type	Description
{flavor_id}	UUID	The UUID of the flavor.

This operation does not accept a request body.

15.2.3.2. Response

Example 15.6. Show flavor details: JSON response

```
{
  "flavor": {
    "description": "",
    "enabled": true,
    "service_profiles": [],
    "service_type": "LOADBALANCERV2",
    "id": "f7b14d9a-b0dc-4fbe-bb14-a0f4970a69e0",
    "name": "dummy"
  }
}
```

15.2.4. Update flavor

Method	URI	Description
PUT	/v2.0/flavors/{flavor_id}	Updates a flavor.

The service_type cannot be updated as there may be associated service profiles and consumers depending on the value.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

15.2.4.1. Request

This table shows the URI parameters for the update flavor request:

Name	Type	Description
{flavor_id}	UUID	The UUID of the flavor.

Example 15.7. Update flavor: JSON request

```
{
  "flavor": {
    "enabled": false,
    "name": "newname",
    "description": "New description"
  }
}
```

15.2.4.2. Response

Example 15.8. Update flavor: JSON response

```
{
  "flavor": {
    "description": "New description",
    "enabled": false,
    "service_profiles": [],
    "service_type": "LOADBALANCERV2",
    "id": "7fc0581b-4509-49e1-90eb-c953c877fa4c",
    "name": "newname"
  }
}
```

15.2.5. Delete flavor

Method	URI	Description
DELETE	/v2.0/flavors/{flavor_id}	Deletes a flavor.

Normal response codes: 204

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.2.5.1. Request

This table shows the URI parameters for the delete flavor request:

Name	Type	Description
{flavor_id}	UUID	The UUID of the flavor.

This operation does not accept a request body.

15.2.6. Associate flavor

Method	URI	Description
POST	/v2.0/flavors/{flavor_id}/service_profiles	Associate a flavor with a service profile.

A flavor can be associated with more than one profile.

Will return 409 Conflict if association already exists.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), conflict (409)

15.2.6.1. Request

This table shows the URI parameters for the associate flavor request:

Name	Type	Description
{flavor_id}	UUID	The UUID of the flavor.

Example 15.9. Associate flavor: JSON request

```
{
  "service_profile": {
    "id": "4e5b9191-ffbe-4f7a-b112-2db98232fd32"
  }
}
```

15.2.6.2. Response

Example 15.10. Associate flavor: JSON response

```
{
  "service_profile": {
    "id": "7fc0581b-4509-49e1-90eb-c953c877fa4c"
  }
}
```

15.2.7. Disassociate a flavor.

Method	URI	Description
DELETE	/v2.0/flavors/{flavor_id}/ service_profiles/{profile_id}	Disassociate a flavor from a service profile.

Normal response codes: 204

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.2.7.1. Request

This table shows the URI parameters for the disassociate a flavor. request:

Name	Type	Description
{flavor_id}	UUID	The UUID of the flavor.
{profile_id}	UUID	The UUID of the profile.

This operation does not accept a request body.

15.2.8. List service profiles

Method	URI	Description
GET	/v2.0/service_profiles	Lists all service profiles visible for the tenant account.

The list can be empty.

Standard query parameters are supported on the URI.

Normal response codes: 200

Error response codes: unauthorized (401)

15.2.8.1. Request

This operation does not accept a request body.

15.2.8.2. Response

Example 15.11. List service profiles: JSON response

```
{
  "service_profiles": [
    {
      "id": "4e5b9191-ffbe-4f7a-b112-2db98232fd32",
      "enabled": true,
      "driver": "neutron_lbaas.drivers.octavia.driver.OctaviaDriver",
      "description": "",
      "metainfo": ""
    },
    {
      "id": "684322c5-703a-48a2-8138-34b99942a7ef",
      "enabled": true,
      "driver": "neutron_lbaas.drivers.octavia.driver.OctaviaDriver",
      "description": "",
      "metainfo": ""
    }
  ]
}
```


15.2.9. Create service profile

Method	URI	Description
POST	/v2.0/service_profiles	Creates a service profile.

This operation establishes a new service profile that can be associated with one or more flavors.

Either metadata or a driver is required.

If a driver is specified but does not exist, call will return a `NotFound` 404 error with the response body explaining that the driver could not be found.

Creation currently limited to administrators. Other users will receive a `Forbidden` 403 response code with a response body `NeutronError` message expressing that creation is disallowed by policy.

If the API cannot fulfill the request due to insufficient data or data that is not valid, the service returns the `HTTP Bad Request` (400) response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.

Normal response codes: 201

Error response codes: `badRequest` (400), `unauthorized` (401), `forbidden` (403)

15.2.9.1. Request

Example 15.12. Create service profile: JSON request

```
{
  "service_profile": {
    "enabled": "true",
    "driver": "neutron_lbaas.drivers.octavia.driver.OctaviaDriver",
    "description": "Dummy profile",
    "metainfo": "{ 'foo': 'bar' }"
  }
}
```

15.2.9.2. Response

Example 15.13. Create service profile: JSON response

```
{
  "service_profile": {
    "enabled": true,
    "metainfo": "{ 'foo': 'bar' }",
    "driver": "neutron_lbaas.drivers.octavia.driver.OctaviaDriver",
    "id": "7c793e5f-9b64-44e0-8b1f-902e59c85a01",
    "description": "Dummy profile"
  }
}
```

15.2.10. Show service profile details

Method	URI	Description
GET	/v2.0/service_profiles/ {profile_id}	Shows details for a service profile.

This operation returns a service profile object by ID. If you are not an administrative user and the object is not visible to your tenant account, the service returns the HTTP Forbidden (403) response code.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.2.10.1. Request

This operation does not accept a request body.

15.2.10.2. Response

Example 15.14. Show service profile details: JSON response

```
{
  "service_profile": {
    "enabled": true,
    "metainfo": "{ 'foo': 'bar' }",
    "driver": "neutron_lbaas.drivers.octavia.driver.OctaviaDriver",
    "id": "7c793e5f-9b64-44e0-8b1f-902e59c85a01",
    "description": "Dummy profile"
  }
}
```

15.2.11. Update service profile

Method	URI	Description
PUT	/v2.0/service_profiles/{profile_id}	Updates a service profile.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

15.2.11.1. Request

Example 15.15. Update service profile: JSON request

```
{
  "service_profile": {
    "enabled": false,
    "driver": "neutron_lbaas.drivers.octavia.driver.OctaviaDriver",
    "description": "New description",
    "metainfo": "{ 'new': 'info' }"
  }
}
```

15.2.11.2. Response

Example 15.16. Update service profile: JSON response

```
{
  "service_profile": {
    "enabled": false,
    "metainfo": "{ 'new': 'info' }",
    "driver": "neutron_lbaas.drivers.octavia.driver.OctaviaDriver",
    "id": "7c793e5f-9b64-44e0-8b1f-902e59c85a01",
    "description": "New description"
  }
}
```

15.2.12. Delete service profile

Method	URI	Description
DELETE	/v2.0/service_profiles/{profile_id}	Deletes a service profile.

Attempting to delete a service profile that is currently associated with a flavor will return a **Conflict 409** with a response body containing an in use message.

Normal response codes: 204

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404), conflict (409)

15.2.12.1. Request

This operation does not accept a request body.

15.3. FWaaS v2.0 (CURRENT) (fw, firewalls, firewall_policies, firewall_rules)

Use the Firewall-as-a-Service (FWaaS) v2.0 extension to deploy firewalls to protect your networks.

The FWaaS extension enables you to:

- Apply firewall rules on traffic entering and leaving tenant networks.
- Apply TCP, UDP, ICMP, or protocol-agnostic rules.
- Create and share firewall policies that hold an ordered collection of the firewall rules.
- Audit firewall rules and policies.

This extension introduces these resources:

- **firewall**. A logical firewall resource that a tenant can instantiate and manage. A firewall can have one firewall policy.
- **firewall_policy**. An ordered collection of firewall rules. You can share a firewall policy across tenants. You can include a firewall policy as part of an audit workflow so that an authorized relevant entity can audit the firewall policy. This entity can differ from the tenant who created, or the tenants that use, the firewall policy.
- **firewall_rule**. A collection of attributes, such as ports and IP addresses. These attributes define match criteria and an action to take, such as allow or deny, on matched data traffic.

Method	URI	Description
GET	/v2.0/fw/firewalls	Lists all firewalls.

Method	URI	Description
POST	/v2.0/fw/firewalls	Creates a firewall.
GET	/v2.0/fw/firewalls/{firewall_id}	Shows details for a firewall.
PUT	/v2.0/fw/firewalls/{firewall_id}	Updates a firewall.
DELETE	/v2.0/fw/firewalls/{firewall_id}	Deletes a firewall.
GET	/v2.0/fw/firewall_policies	Lists all firewall policies.
POST	/v2.0/fw/firewall_policies	Creates a firewall policy.
GET	/v2.0/fw/firewall_policies/{firewall_policy_id}	Shows details for a firewall policy.
PUT	/v2.0/fw/firewall_policies/{firewall_policy_id}	Updates a firewall policy.
DELETE	/v2.0/fw/firewall_policies/{firewall_policy_id}	Deletes a firewall policy.
PUT	/v2.0/fw/firewall_policies/{firewall_policy_id}/insert_rule	Insert firewall rule into a policy.
PUT	/v2.0/fw/firewall_policies/{firewall_policy_id}/remove_rule	Remove firewall rule from a policy.
GET	/v2.0/fw/firewall_rules	Lists all firewall rules.
POST	/v2.0/fw/firewall_rules	Creates a firewall rule.
GET	/v2.0/fw/firewall_rules/{firewall_rule_id}	Shows details for a firewall rule.
PUT	/v2.0/fw/firewall_rules/{firewall_rule_id}	Updates a firewall rule.
DELETE	/v2.0/fw/firewall_rules/{firewall_rule_id}	Deletes a firewall rule.

15.3.1. List firewalls

Method	URI	Description
GET	/v2.0/fw/firewalls	Lists all firewalls.

The list might be empty.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

15.3.1.1. Request

This operation does not accept a request body.

15.3.1.2. Response

Example 15.17. List firewalls: JSON response

```
{
  "firewalls": [
    {
      "admin_state_up": true,
      "description": "",
      "firewall_policy_id": "c69933c1-b472-44f9-8226-30dc4ffd454c",
      "id": "3b0ef8f4-82c7-44d4-a4fb-6177f9a21977",
      "name": "",
      "status": "ACTIVE",
      "router_ids": [
        "650bfd2f-7766-4a0d-839f-218f33e16998"
      ],
      "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
    }
  ]
}
```

15.3.2. Create firewall

Method	URI	Description
POST	/v2.0/fw/firewalls	Creates a firewall.

The firewall must be associated with a firewall policy.

If `admin_state_up` is `false`, the firewall would block all traffic.

Normal response codes: 201

Error response codes: `badRequest` (400), `unauthorized` (401)

15.3.2.1. Request

Example 15.18. Create firewall: JSON request

```
{
  "firewall": {
    "admin_state_up": true,
    "firewall_policy_id": "c69933c1-b472-44f9-8226-30dc4ffd454c"
  }
}
```

15.3.2.2. Response

Example 15.19. Create firewall: JSON response

```
{
  "firewall": {
    "admin_state_up": true,
    "description": "",
    "firewall_policy_id": "c69933c1-b472-44f9-8226-30dc4ffd454c",
    "id": "3b0ef8f4-82c7-44d4-a4fb-6177f9a21977",
    "name": "",
    "status": "PENDING_CREATE",
    "router_ids": [
      "650bfd2f-7766-4a0d-839f-218f33e16998"
    ],
    "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
  }
}
```

15.3.3. Show firewall details

Method	URI	Description
GET	/v2.0/fw/firewalls/{firewall_id}	Shows details for a firewall.

If the user is not an administrative user and the firewall object does not belong to the tenant account, this call returns the Forbidden (403) response code.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.3.3.1. Request

This table shows the URI parameters for the show firewall details request:

Name	Type	Description
{firewall_id}	UUID	The UUID of the firewall.

This operation does not accept a request body.

15.3.3.2. Response

Example 15.20. Show firewall details: JSON response

```
{
  "firewall": {
    "admin_state_up": true,
    "description": "",
    "firewall_policy_id": "c69933c1-b472-44f9-8226-30dc4ffd454c",
    "id": "3b0ef8f4-82c7-44d4-a4fb-6177f9a21977",
    "name": "",
    "status": "ACTIVE",
    "router_ids": [
      "650bfd2f-7766-4a0d-839f-218f33e16998"
    ],
    "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
  }
}
```


15.3.4. Update firewall

Method	URI	Description
PUT	/v2.0/fw/firewalls/{firewall_id}	Updates a firewall.

To update a service, the service status cannot be a PENDING_* status.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.3.4.1. Request

This table shows the URI parameters for the update firewall request:

Name	Type	Description
{firewall_id}	UUID	The UUID of the firewall.

Example 15.21. Update firewall: JSON request

```
{
  "firewall": {
    "admin_state_up": "false"
  }
}
```

15.3.4.2. Response

Example 15.22. Update firewall: JSON response

```
{
  "firewall": {
    "admin_state_up": false,
    "description": "",
    "firewall_policy_id": "c69933c1-b472-44f9-8226-30dc4ffd454c",
    "id": "3b0ef8f4-82c7-44d4-a4fb-6177f9a21977",
    "name": "",
    "status": "PENDING_UPDATE",
    "router_ids": [
      "650bfd2f-7766-4a0d-839f-218f33e16998"
    ],
    "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
  }
}
```

15.3.5. Delete firewall

Method	URI	Description
DELETE	/v2.0/fw/firewalls/{firewall_id}	Deletes a firewall.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.3.5.1. Request

This table shows the URI parameters for the delete firewall request:

Name	Type	Description
{firewall_id}	UUID	The UUID of the firewall.

This operation does not accept a request body.

15.3.6. List firewall policies

Method	URI	Description
GET	/v2.0/fw/firewall_policies	Lists all firewall policies.

The list might be empty.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

15.3.6.1. Request

This operation does not accept a request body.

15.3.6.2. Response

Example 15.23. List firewall policies: JSON response

```
{
  "firewall_policies": [
    {
      "audited": false,
      "description": "",
      "firewall_rules": [
        "8722e0e0-9cc9-4490-9660-8c9a5732fbb0"
      ],
      "id": "c69933c1-b472-44f9-8226-30dc4ffd454c",
      "name": "test-policy",
      "shared": false,
      "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
    }
  ]
}
```

15.3.7. Create firewall policy

Method	URI	Description
POST	/v2.0/fw/firewall_policies	Creates a firewall policy.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

15.3.7.1. Request

Example 15.24. Create firewall policy: JSON request

```
{
  "firewall_policy": {
    "firewall_rules": [
      "8722e0e0-9cc9-4490-9660-8c9a5732fbb0"
    ],
    "name": "test-policy"
  }
}
```

15.3.7.2. Response

Example 15.25. Create firewall policy: JSON response

```
{
  "firewall_policy": {
    "audited": false,
    "description": "",
    "firewall_rules": [
      "8722e0e0-9cc9-4490-9660-8c9a5732fbb0"
    ],
    "id": "c69933c1-b472-44f9-8226-30dc4ffd454c",
    "name": "test-policy",
    "shared": false,
    "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
  }
}
```

15.3.8. Show firewall policy details

Method	URI	Description
GET	/v2.0/fw/firewall_policies/{firewall_policy_id}	Shows details for a firewall policy.

If the user is not an administrative user and the firewall policy object does not belong to the tenant account, this call returns the Forbidden (403) response code.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.3.8.1. Request

This table shows the URI parameters for the show firewall policy details request:

Name	Type	Description
{firewall_policy_id}	UUID	The UUID of the firewall policy.

This operation does not accept a request body.

15.3.8.2. Response

Example 15.26. Show firewall policy details: JSON response

```
{
  "firewall_policy": {
    "audited": false,
    "description": "",
    "firewall_rules": [
      "8722e0e0-9cc9-4490-9660-8c9a5732fbb0"
    ],
    "id": "c69933c1-b472-44f9-8226-30dc4ffd454c",
    "name": "test-policy",
    "shared": false,
    "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
  }
}
```

15.3.9. Update firewall policy

Method	URI	Description
PUT	/v2.0/fw/firewall_policies/{firewall_policy_id}	Updates a firewall policy.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.3.9.1. Request

This table shows the URI parameters for the update firewall policy request:

Name	Type	Description
{firewall_policy_id}	UUID	The UUID of the firewall policy.

Example 15.27. Update firewall policy: JSON request

```
{
  "firewall_policy": {
    "firewall_rules": [
      "a08ef905-0ff6-4784-8374-175fffe7dade",
      "8722e0e0-9cc9-4490-9660-8c9a5732fbb0"
    ]
  }
}
```

15.3.9.2. Response

Example 15.28. Update firewall policy: JSON response

```
{
  "firewall_policy": {
    "audited": false,
    "description": "",
    "firewall_rules": [
      "a08ef905-0ff6-4784-8374-175fffe7dade",
      "8722e0e0-9cc9-4490-9660-8c9a5732fbb0"
    ],
    "id": "c69933c1-b472-44f9-8226-30dc4fffd454c",
    "name": "test-policy",
    "shared": false,
    "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
  }
}
```

15.3.10. Delete firewall policy

Method	URI	Description
DELETE	/v2.0/fw/firewall_policies/{firewall_policy_id}	Deletes a firewall policy.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.3.10.1. Request

This table shows the URI parameters for the delete firewall policy request:

Name	Type	Description
{firewall_policy_id}	UUID	The UUID of the firewall policy.

This operation does not accept a request body.

15.3.11. Insert rule into a firewall policy

Method	URI	Description
PUT	/v2.0/fw/firewall_policies/{firewall_policy_id}/insert_rule	Insert firewall rule into a policy.

A `firewall_rule_id` is inserted relative to the position of the `firewall_rule_id` set in `insert_before` or `insert_after`. If `insert_before` is set, `insert_after` is ignored. If both `insert_before` and `insert_after` are not set, the new `firewall_rule_id` is inserted at the top of the policy.

Normal response codes: 200

Error response codes: `badRequest` (400), `unauthorized` (401), `itemNotFound` (404), `conflict` (409)

15.3.11.1. Request

This table shows the URI parameters for the insert rule into a firewall policy request:

Name	Type	Description
{firewall_policy_id}	UUID	The UUID of the firewall policy.

Example 15.29. Insert rule into a firewall policy: JSON request

```
{
  "firewall_rule_id": "7bc34b8c-8d3b-4ada-a9c8-1f4c11c65692",
  "insert_after": "a08ef905-0ff6-4784-8374-175fffe7dade",
  "insert_before": ""
}
```

15.3.11.2. Response

Example 15.30. Insert rule into a firewall policy: JSON response

```
{
  "audited": false,
  "description": "",
  "firewall_list": [],
  "firewall_rules": [
    "a08ef905-0ff6-4784-8374-175fffe7dade",
    "7bc34b8c-8d3b-4ada-a9c8-1f4c11c65692",
    "8722e0e0-9cc9-4490-9660-8c9a5732fbb0"
  ],
  "id": "c69933c1-b472-44f9-8226-30dc4ffd454c",
  "name": "test-policy",
  "shared": false,
  "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
}
```


15.3.12. Remove rule from firewall policy

Method	URI	Description
PUT	/v2.0/fw/firewall_policies/{firewall_policy_id}/remove_rule	Remove firewall rule from a policy.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.3.12.1. Request

This table shows the URI parameters for the remove rule from firewall policy request:

Name	Type	Description
{firewall_policy_id}	UUID	The UUID of the firewall policy.

Example 15.31. Remove rule from firewall policy: JSON request

```
{
  "firewall_rule_id": "7bc34b8c-8d3b-4ada-a9c8-1f4c11c65692"
}
```

15.3.12.2. Response

Example 15.32. Remove rule from firewall policy: JSON response

```
{
  "audited": false,
  "description": "",
  "firewall_list": [],
  "firewall_rules": [
    "a08ef905-0ff6-4784-8374-175fffe7dade",
    "8722e0e0-9cc9-4490-9660-8c9a5732fbb0"
  ],
  "id": "c69933c1-b472-44f9-8226-30dc4ffd454c",
  "name": "test-policy",
  "shared": false,
  "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
}
```

15.3.13. List firewall rules

Method	URI	Description
GET	/v2.0/fw/firewall_rules	Lists all firewall rules.

The list might be empty.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

15.3.13.1. Request

This operation does not accept a request body.

15.3.13.2. Response

Example 15.33. List firewall rules: JSON response

```
{
  "firewall_rules": [
    {
      "action": "allow",
      "description": "",
      "destination_ip_address": null,
      "destination_port": "80",
      "enabled": true,
      "firewall_policy_id": "c69933c1-b472-44f9-8226-30dc4ffd454c",
      "id": "8722e0e0-9cc9-4490-9660-8c9a5732fbb0",
      "ip_version": 4,
      "name": "ALLOW_HTTP",
      "position": 1,
      "protocol": "tcp",
      "shared": false,
      "source_ip_address": null,
      "source_port": null,
      "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
    }
  ]
}
```

15.3.14. Create firewall rule

Method	URI	Description
POST	/v2.0/fw/firewall_rules	Creates a firewall rule.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

15.3.14.1. Request

Example 15.34. Create firewall rule: JSON request

```
{
  "firewall_rule": {
    "action": "allow",
    "destination_port": "80",
    "enabled": true,
    "name": "ALLOW_HTTP",
    "protocol": "tcp"
  }
}
```

15.3.14.2. Response

Example 15.35. Create firewall rule: JSON response

```
{
  "firewall_rule": {
    "action": "allow",
    "description": "",
    "destination_ip_address": null,
    "destination_port": "80",
    "enabled": true,
    "firewall_policy_id": null,
    "id": "8722e0e0-9cc9-4490-9660-8c9a5732fbb0",
    "ip_version": 4,
    "name": "ALLOW_HTTP",
    "position": null,
    "protocol": "tcp",
    "shared": false,
    "source_ip_address": null,
    "source_port": null,
    "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
  }
}
```

15.3.15. Show firewall rule details

Method	URI	Description
GET	/v2.0/fw/firewall_rules/{firewall_rule_id}	Shows details for a firewall rule.

If the user is not an administrative user and the firewall rule object does not belong to the tenant account, this call returns the Forbidden (403) response code.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.3.15.1. Request

This table shows the URI parameters for the show firewall rule details request:

Name	Type	Description
{firewall_rule_id}	UUID	The UUID for the firewall rule.

This operation does not accept a request body.

15.3.15.2. Response

Example 15.36. Show firewall rule details: JSON response

```
{
  "firewall_rule": {
    "action": "allow",
    "description": "",
    "destination_ip_address": null,
    "destination_port": "80",
    "enabled": true,
    "firewall_policy_id": null,
    "id": "8722e0e0-9cc9-4490-9660-8c9a5732fbb0",
    "ip_version": 4,
    "name": "ALLOW_HTTP",
    "position": null,
    "protocol": "tcp",
    "shared": false,
    "source_ip_address": null,
    "source_port": null,
    "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
  }
}
```

15.3.16. Update firewall rule

Method	URI	Description
PUT	/v2.0/fw/firewall_rules/{firewall_rule_id}	Updates a firewall rule.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.3.16.1. Request

This table shows the URI parameters for the update firewall rule request:

Name	Type	Description
{firewall_rule_id}	UUID	The UUID for the firewall rule.

Example 15.37. Update firewall rule: JSON request

```
{
  "firewall_rule": {
    "shared": "true"
  }
}
```

15.3.16.2. Response

Example 15.38. Update firewall rule: JSON response

```
{
  "firewall_rule": {
    "action": "allow",
    "description": "",
    "destination_ip_address": null,
    "destination_port": "80",
    "enabled": true,
    "firewall_policy_id": "c69933c1-b472-44f9-8226-30dc4ffd454c",
    "id": "8722e0e0-9cc9-4490-9660-8c9a5732fbb0",
    "ip_version": 4,
    "name": "ALLOW_HTTP",
    "position": 1,
    "protocol": "tcp",
    "shared": true,
    "source_ip_address": null,
    "source_port": null,
    "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
  }
}
```

15.3.17. Delete firewall rule

Method	URI	Description
DELETE	/v2.0/fw/firewall_rules/{firewall_rule_id}	Deletes a firewall rule.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.3.17.1. Request

This table shows the URI parameters for the delete firewall rule request:

Name	Type	Description
{firewall_rule_id}	UUID	The UUID for the firewall rule.

This operation does not accept a request body.

15.4. LBaaS 2.0 (STABLE) (Ibaas, loadbalancers, listeners, health_monitors, pools, members)

The Load-Balancer-as-a-Service (LBaaS) version 2.0 extension pairs with the Networking 2.0 API to enable OpenStack tenants to manage load balancers for their VMs. With this extension you can load-balance client traffic from one network to application services, such as VMs, on the same network.

Use this extension to create and manage load balancers, listeners, pools, members of a pool, and health monitors and view status of a resource.

Table 15.1. Load balancer statuses

Status	Description
ACTIVE	The resource is ready and active.
PENDING_CREATE	The resource is being created.
PENDING_UPDATE	The resource is being updated.
PENDING_DELETE	The resource is pending deletion.
INACTIVE	The resource is not active.
ERROR	An object within the service is not working. The <code>error_details</code> attribute provides an explanation for the error, its cause, and possibly a solution.

Method	URI	Description
GET	/v2.0/lbaas/loadbalancers	Lists all load balancers for the tenant account.
POST	/v2.0/lbaas/loadbalancers	Creates a load balancer.
GET	/v2.0/lbaas/loadbalancers/{loadbalancer_id}	Shows details for a load balancer.
PUT	/v2.0/lbaas/loadbalancers/{loadbalancer_id}	Updates a load balancer.

Method	URI	Description
DELETE	/v2.0/lbaas/loadbalancers/{loadbalancer_id}	Removes a load balancer and its associated configuration from the tenant account.
GET	/v2.0/lbaas/loadbalancers/{loadbalancer_id}/statuses	Shows the status tree for a load balancer.
GET	/v2.0/lbaas/listeners	Lists all listeners.
POST	/v2.0/lbaas/listeners	Creates a listener.
GET	/v2.0/lbaas/listeners/{listener_id}	Shows details for a listener.
PUT	/v2.0/lbaas/listeners/{listener_id}	Updates a listener.
DELETE	/v2.0/lbaas/listeners/{listener_id}	Removes a listener.
GET	/v2.0/lbaas/pools	Lists all pools that are associated with your tenant account.
POST	/v2.0/lbaas/pools	Creates a pool.
GET	/v2.0/lbaas/pools/{pool_id}	Shows details for a pool.
PUT	/v2.0/lbaas/pools/{pool_id}	Updates a pool.
DELETE	/v2.0/lbaas/pools/{pool_id}	Removes a pool.
GET	/v2.0/lbaas/pools/{pool_id}/members	Lists members of a pool.
POST	/v2.0/lbaas/pools/{pool_id}/members	Adds a member to a pool.
GET	/v2.0/lbaas/pools/{pool_id}/members/{member_id}	Shows details for a pool member.
PUT	/v2.0/lbaas/pools/{pool_id}/members/{member_id}	Updates attributes for a pool member.
DELETE	/v2.0/lbaas/pools/{pool_id}/members/{member_id}	Removes a member from a pool and its associated configuration from the tenant account.
POST	/v2.0/lbaas/health_monitors	Creates a health monitor.
GET	/v2.0/lbaas/health_monitors	Lists health monitors.
GET	/v2.0/lbaas/health_monitors/{health_monitor_id}	Shows details for a health monitor.
PUT	/v2.0/lbaas/health_monitors/{health_monitor_id}	Updates a health monitor.
DELETE	/v2.0/lbaas/health_monitors/{health_monitor_id}	Removes a health monitor and its associated configuration from the tenant account.

15.4.1. List load balancers

Method	URI	Description
GET	/v2.0/lbaas/loadbalancers	Lists all load balancers for the tenant account.

The list might be empty.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404), Internal-server-error (500), serviceUnavailable (503)

15.4.1.1. Request

This operation does not accept a request body.

15.4.1.2. Response

Example 15.39. List load balancers: JSON response

```
{
  "loadbalancers": [
    {
      "description": "simple lb",
      "admin_state_up": true,
      "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
      "provisioning_status": "ACTIVE",
      "listeners": [],
      "vip_address": "10.0.0.2",
      "vip_subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
      "id": "a9729389-6147-41a3-ab22-a24aed8692b2",
      "operating_status": "ONLINE",
      "name": "loadbalancer1"
    }
  ]
}
```


15.4.2. Create load balancer

Method	URI	Description
POST	/v2.0/lbaas/loadbalancers	Creates a load balancer.

This operation provisions a new load balancer by using the configuration that you define in the request object. After the API validates the request and starts the provisioning process, the API returns a response object that contains a unique ID and the status of provisioning the load balancer.

In the response, the load balancer provisioning status is ACTIVE, PENDING_CREATE, or ERROR.

If the status is PENDING_CREATE, issue **GET** /lbaas/loadbalancers/loadbalancer_id to view the progress of the provisioning operation. When the load balancer status changes to ACTIVE, the load balancer is successfully provisioned and operational for traffic handling.

If the API cannot fulfill the request due to insufficient data or data that is not valid, the service returns the HTTP Bad Request (400) response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.

You can configure all documented features of the load balancer at creation time by specifying the additional elements or attributes in the request.

Administrative users can specify a tenant ID that is different than their own to create load balancers for other tenants.

Example: Create a load balancer

- `tenant_id`. Admin only. Required to create a load balancer for another tenant.
- `vip_subnet_id`. The network on which to allocate the VIP address for the load balancer. A tenant can only create load balancer VIPs on networks that the policy authorizes, such as her own networks or shared or provider networks.

Some attributes receive default values if you omit them from the request:

- `admin_state_up`. Default is true.
- `name`. Default is an empty string.
- `description`. Default is an empty string.

If the API cannot fulfill the request due to insufficient data or data that is not valid, the service returns the HTTP Bad Request (400) response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.

You can configure all documented features of the load balancer at creation time by specifying the additional elements or attributes in the request.

Administrative users can specify a tenant ID that is different than their own to create load balancers for other tenants.

If you own the subnet where you want to create the load balancer VIP, you can specify a `vip_address` attribute. If you omit the `vip_address` attribute from the payload, the LBaaS service allocates a VIP address from the subnet of the load balancer VIP.

An optional `flavor` attribute can be passed to enable dynamic selection of an appropriate provider if configured by the operator. The basic selection algorithm chooses the provider in the first service profile currently associated with flavor.

You can also specify the `provider` attribute when you create a load balancer. You can set this attribute to any service provider with a `LOADBALANCER` service type. Setting both a flavor and a provider will result in a conflict error.

Example: Create a load balancer

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.2.1. Request

Example 15.40. Create load balancer: JSON request

```
{
  "loadbalancer": {
    "name": "loadbalancer1",
    "description": "simple lb",
    "tenant_id": "b7c1a69e88bf4b21a8148f787aef2081",
    "vip_subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
    "vip_address": "10.0.0.4",
    "admin_state_up": true,
    "flavor": "a7ae5d5a-d855-4f9a-b187-af66b53f4d04"
  }
}
```

15.4.2.2. Response

Example 15.41. Create load balancer: JSON response

```
{
  "loadbalancer": {
    "admin_state_up": true,
    "description": "simple lb",
    "id": "a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "listeners": [],
    "name": "loadbalancer1",
    "operating_status": "ONLINE",
    "provisioning_status": "ACTIVE",
    "tenant_id": "b7c1a69e88bf4b21a8148f787aef2081",
    "vip_address": "10.0.0.4",
    "vip_subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
    "flavor": "a7ae5d5a-d855-4f9a-b187-af66b53f4d04",
    "provider": "sample_provider"
  }
}
```

```
}  
}
```

15.4.3. Show load balancer details

Method	URI	Description
GET	/v2.0/lbaas/loadbalancers/{loadbalancer_id}	Shows details for a load balancer.

This operation returns a load balancer object, by ID. If you are not an administrative user and the load balancer object does not belong to your tenant account, the service returns the HTTP Forbidden (403) response code.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.3.1. Request

This operation does not accept a request body.

15.4.3.2. Response

Example 15.42. Show load balancer details: JSON response

```
{
  "loadbalancer": {
    "description": "simple lb",
    "admin_state_up": true,
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
    "provisioning_status": "ACTIVE",
    "listeners": [],
    "vip_address": "10.0.0.2",
    "vip_subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
    "id": "a9729389-6147-41a3-ab22-a24aed8692b2",
    "operating_status": "ONLINE",
    "name": "loadbalancer1"
  }
}
```

15.4.4. Update load balancer

Method	URI	Description
PUT	/v2.0/lbaas/loadbalancers/{loadbalancer_id}	Updates a load balancer.

If the request is valid, the service returns the Accepted (202) response code. To confirm the update, check that the load balancer provisioning status is ACTIVE. If the status is PENDING_UPDATE, use a GET operation to poll the load balancer object for changes.

This operation returns the updated load balancer object with the ACTIVE, PENDING_UPDATE, or ERROR provisioning status.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.4.1. Request

Example 15.43. Update load balancer: JSON request

```
{
  "loadbalancer": {
    "admin_state_up": false,
    "description": "simple lb2",
    "name": "loadbalancer2"
  }
}
```

15.4.4.2. Response

Example 15.44. Update load balancer: JSON response

```
{
  "loadbalancer": {
    "admin_state_up": false,
    "description": "simple lb2",
    "id": "a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "listeners": [],
    "name": "loadbalancer2",
    "operating_status": "ONLINE",
    "provisioning_status": "PENDING_UPDATE",
    "tenant_id": "b7c1a69e88bf4b21a8148f787aef2081",
    "vip_address": "10.0.0.4",
    "vip_subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2"
  }
}
```

15.4.5. Remove load balancer

Method	URI	Description
DELETE	/v2.0/lbaas/loadbalancers/{loadbalancer_id}	Removes a load balancer and its associated configuration from the tenant account.

The API immediately purges any and all configuration data. You cannot recover it.

Example: Delete a load balancer

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.5.1. Request

This operation does not accept a request body.

15.4.6. Show load balancer status tree

Method	URI	Description
GET	/v2.0/lbaas/loadbalancers/{loadbalancer_id}/statuses	Shows the status tree for a load balancer.

This operation returns a status tree for a load balancer object, by load balancer ID. If you are not an administrative user and the load balancer object does not belong to the tenant account, the API returns the Forbidden (403) response code.

If the operation succeeds, the returned element is a status tree that contains the load balancer and all provisioning and operating statuses for its children.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.6.1. Request

This operation does not accept a request body.

15.4.6.2. Response

Example 15.45. Show load balancer status tree: JSON response

```
{
  "statuses": {
    "loadbalancer": {
      "name": "lb1",
      "listeners": [
        {
          "pools": [
            {
              "name": "pool1",
              "provisioning_status": "ACTIVE",
              "health_monitor": {
                "type": "HTTP",
                "id": "90f7c765-0bc9-47c4-8513-4cc0c264c8f8",
                "provisioning_status": "ACTIVE"
              },
              "members": [
                {
                  "address": "10.0.0.4",
                  "protocol_port": 80,
                  "id": "32723bee-2484-4de3-b6fc-c0b98d35fc84",
                  "operating_status": "ONLINE",
                  "provisioning_status": "ACTIVE"
                },
                {
                  "address": "10.0.0.3",
                  "protocol_port": 80,
                  "id": "173b8164-0c9a-43ec-ab33-4ae0e7a8f863",
```

```
        "operating_status": "ONLINE",
        "provisioning_status": "ACTIVE"
      },
      ],
      "id": "ae6f93b8-a3f6-46cd-bb18-c2ab0308abf7",
      "operating_status": "ONLINE"
    }
  ],
  "name": "listener1",
  "id": "c2a41fbe-b70a-4645-bb11-4d3c28f23a25",
  "operating_status": "ONLINE",
  "provisioning_status": "ACTIVE"
},
{
  "id": "a4c19566-6f81-4c96-ac11-33954a9825a2",
  "operating_status": "ONLINE",
  "provisioning_status": "ACTIVE"
}
}
```


15.4.7. List listeners

Method	URI	Description
GET	/v2.0/lbaas/listeners	Lists all listeners.

This operation lists all listeners that are associated with your tenant account.

The list might be empty.

Example: List listeners

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

15.4.7.1. Request

This operation does not accept a request body.

15.4.7.2. Response

Example 15.46. List listeners: JSON response

```
{
  "listeners": [
    {
      "admin_state_up": true,
      "connection_limit": 100,
      "default_pool_id": null,
      "description": "",
      "id": "35cb8516-1173-4035-8dae-0dae3453f37f",
      "loadbalancers": [
        {
          "id": "a9729389-6147-41a3-ab22-a24aed8692b2"
        }
      ],
      "name": "",
      "protocol": "HTTP",
      "protocol_port": 80,
      "tenant_id": "3e4d8bec50a845fcb09e03a4375c691d",
      "default_tls_container_ref": "https://barbican.endpoint/containers/a36c20d0-18e9-42ce-88fd-82a35977ee8c",
      "sni_container_refs": [
        "https://barbican.endpoint/containers/b36c20d0-18e9-42ce-88fd-82a35977ee8d",
        "https://barbican.endpoint/containers/c36c20d0-18e9-42ce-88fd-82a35977ee8e"
      ]
    }
  ]
}
```

15.4.8. Create listener

Method	URI	Description
POST	/v2.0/lbaas/listeners	Creates a listener.

This operation provisions a new listener by using the configuration that you define in the request object. After the request is validated and the provisioning process begins, a response object is returned. The object contains a unique identifier.

At a minimum, you must specify these listener attributes:

- `tenant_id`. Admin only. Required to create a listener for another tenant.
- `loadbalancer_id`. The load balancer on which to provision this listener. A tenant can only create listeners on load balancers that the policy authorizes. For example, her own load balancers.
- `description`. The load balancer description.
- `protocol`. The protocol for which the front end listens. Must be HTTP, HTTPS, TCP, or TERMINATED_HTTPS.

Some attributes receive default values if you omit them from the request:

- `protocol_port`. The port on which the front end listens. Must be an integer from 1 to 65535.
- `default_tls_container_ref`. The reference to a container that holds TLS secrets. If you also specify `sni_container_refs`, this container is the default. This parameter is required for the TERMINATED_HTTPS protocol.
- `sni_container_refs`. A list of references to containers that hold TLS secrets for server name indication (SNI). This parameter is required for the TERMINATED_HTTPS protocol.
- `admin_state_up`. Default is `true`.
- `name`. Default is an empty string.
- `description`. Default is an empty string.
- `connection_limit`. Default is `-1`, which indicates an infinite limit.

If the API cannot fulfill the request due to insufficient data or data that is not valid, the service returns the HTTP Bad Request (400) response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.

You can configure all documented features of the listener at creation time by specifying the additional elements or attributes in the request.

Administrative users can specify a tenant ID that is different than their own to create listeners for other tenants.

To update a listener, the load balancer to which to attach must have an ACTIVE provisioning status.

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.8.1. Request

Example 15.47. Create listener: JSON request

```
{
  "listener": {
    "admin_state_up": true,
    "connection_limit": 100,
    "description": "listener one",
    "loadbalancer_id": "a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "name": "listener1",
    "protocol": "HTTP",
    "protocol_port": "80",
    "default_tls_container_ref": "https://barbican.endpoint/containers/
a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "sni_container_refs": [
      "https://barbican.endpoint/containers/
b36c20d0-18e9-42ce-88fd-82a35977ee8d",
      "https://barbican.endpoint/containers/
c36c20d0-18e9-42ce-88fd-82a35977ee8e"
    ]
  }
}
```

15.4.8.2. Response

Example 15.48. Create listener: JSON response

```
{
  "listener": {
    "admin_state_up": true,
    "connection_limit": 100,
    "default_pool_id": null,
    "description": "listener one",
    "id": "39de4d56-d663-46e5-85a1-5b9d5fa17829",
    "loadbalancers": [
      {
        "id": "a36c20d0-18e9-42ce-88fd-82a35977ee8c"
      }
    ],
    "name": "listener1",
    "protocol": "HTTP",
    "protocol_port": 80,
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
    "default_tls_container_ref": "https://barbican.endpoint/containers/
a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "sni_container_refs": [
      "https://barbican.endpoint/containers/
b36c20d0-18e9-42ce-88fd-82a35977ee8d",

```

```
        "https://barbican.endpoint/containers/  
c36c20d0-18e9-42ce-88fd-82a35977ee8e"  
    }  
}
```

15.4.9. Show listener details

Method	URI	Description
GET	/v2.0/lbaas/listeners/{listener_id}	Shows details for a listener.

This operation returns a listener object, by ID. If you are not an administrative user and the listener object does not belong to your account, the API returns the HTTP Forbidden (403) response code.

Example: Show listener details

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.9.1. Request

This operation does not accept a request body.

15.4.9.2. Response

Example 15.49. Show listener details: JSON response

```
{
  "listener": {
    "admin_state_up": true,
    "connection_limit": 100,
    "default_pool_id": null,
    "description": "",
    "id": "35cb8516-1173-4035-8dae-0dae3453f37f",
    "loadbalancers": [
      {
        "id": "a9729389-6147-41a3-ab22-a24aed8692b2"
      }
    ],
    "name": "",
    "protocol": "HTTP",
    "protocol_port": 80,
    "tenant_id": "3e4d8bec50a845fcb09e03a4375c691d",
    "default_tls_container_ref": "https://barbican.endpoint/containers/a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "sni_container_refs": [
      "https://barbican.endpoint/containers/b36c20d0-18e9-42ce-88fd-82a35977ee8d",
      "https://barbican.endpoint/containers/c36c20d0-18e9-42ce-88fd-82a35977ee8e"
    ]
  }
}
```

15.4.10. Update listener

Method	URI	Description
PUT	/v2.0/lbaas/listeners/{listener_id}	Updates a listener.

This operation updates the attributes of a listener. Upon successful validation of the request, the service returns the HTTP Accepted (202) response code.

Note: You cannot update the `listener_id`, `tenant_id`, `loadbalancer_id`, `loadbalancers`, `default_pool_id`, `protocol`, and `protocol_port` attributes. Attempting to update an immutable attribute results in the HTTP Immutable (422) response code.

Note: You cannot update a listener if the load balancer to which the listener is attached does not have an ACTIVE provisioning status.

Normal response codes: 200

Error response codes: Internal-server-error (500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

15.4.10.1. Request

Example 15.50. Update listener: JSON request

```
{
  "listener": {
    "admin_state_up": false,
    "connection_limit": 200,
    "description": "listener two",
    "name": "listener2",
    "default_tls_container_ref": "https://barbican.endpoint/containers/a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "sni_container_refs": [
      "https://barbican.endpoint/containers/b36c20d0-18e9-42ce-88fd-82a35977ee8d",
      "https://barbican.endpoint/containers/c36c20d0-18e9-42ce-88fd-82a35977ee8e"
    ]
  }
}
```

15.4.10.2. Response

Example 15.51. Update listener: JSON response

```
{
  "listener": {
    "admin_state_up": false,
    "connection_limit": 200,
    "default_pool_id": null,
    "description": "listener two",
    "id": "39de4d56-d663-46e5-85a1-5b9d5fa17829",
  }
}
```

```
    "loadbalancers": [
      {
        "id": "a36c20d0-18e9-42ce-88fd-82a35977ee8c"
      }
    ],
    "name": "listener2",
    "protocol": "HTTP",
    "protocol_port": 80,
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
    "default_tls_container_ref": "https://barbican.endpoint/containers/
a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "sni_container_refs": [
      "https://barbican.endpoint/containers/
b36c20d0-18e9-42ce-88fd-82a35977ee8d",
      "https://barbican.endpoint/containers/
c36c20d0-18e9-42ce-88fd-82a35977ee8e"
    ]
  }
}
```

15.4.11. Remove listener

Method	URI	Description
DELETE	/v2.0/lbaas/listeners/{listener_id}	Removes a listener.

This operation removes a listener and its associated configuration from the tenant account. The API immediately purges any and all configuration data. You cannot recover it.

You cannot delete a listener if the load balancer to which it is attached does not have an ACTIVE provisioning status.

Example: Delete a listener

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.11.1. Request

This operation does not accept a request body.

15.4.12. List pools

Method	URI	Description
GET	/v2.0/lbaas/pools	Lists all pools that are associated with your tenant account.

The list might be empty.

Example: List pools

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

15.4.12.1. Request

This operation does not accept a request body.

15.4.12.2. Response

Example 15.52. List pools: JSON response

```
{
  "pools": [
    {
      "status": "ACTIVE",
      "lb_algorithm": "ROUND_ROBIN",
      "protocol": "HTTP",
      "description": "",
      "health_monitors": [
        "b7633ade-24dc-4d72-8475-06aa22be5412"
      ],
      "members": [
        "cf024846-7516-4e3a-b0fb-6590322c836f"
      ],
      "status_description": null,
      "id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332",
      "vip_id": "388c739a-6a57-4e74-bc7b-a5cd60248bba",
      "name": "pool1",
      "admin_state_up": true,
      "subnet_id": "aa547115-d710-4d6d-bb2c-b038d9c2704b",
      "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
      "health_monitors_status": [
        {
          "monitor_id": "b7633ade-24dc-4d72-8475-06aa22be5412",
          "status": "ACTIVE",
          "status_description": null
        }
      ],
      "provider": "haproxy"
    }
  ]
}
```

15.4.13. Create pool

Method	URI	Description
POST	/v2.0/lbaas/pools	Creates a pool.

This operation provisions a pool by using the configuration that you define in the request object. After the API validates the request and starts the provisioning process, the API returns a response object, which contains a unique ID.

At a minimum, you must specify these pool attributes:

- `tenant_id`. Admin only. Required to create a pool for another tenant.
- `protocol`. The protocol for which this pool and its members listen. A valid value is TCP, HTTP, or HTTPS.
- `lb_algorithm`. The load-balancer algorithm, such as ROUND_ROBIN, LEAST_CONNECTIONS, and SOURCE_IP, that distributes traffic to the pool members. The load-balancer provider must support this algorithm.
- `listener_id`. The UUID of the listener in which this pool becomes the default pool. Each listener has only one default pool.

Some attributes receive default values if you omit them from the request:

- `admin_state_up`. Default is `true`.
- `name`. Default is an empty string.
- `description`. Default is an empty string.
- `session_persistence`. Default is an empty dictionary.

If the API cannot fulfill the request due to insufficient data or data that is not valid, the service returns the HTTP Bad Request (400) response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.

Users can configure all documented features at creation time by providing the additional elements or attributes in the request.

Administrative users can specify a tenant ID that is different than their own to create pools for other tenants.

To update a pool, the load balancer to which to attach must have an ACTIVE provisioning status.

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.13.1. Request

Example 15.53. Create pool: JSON request

```
{
  "pool": {
    "admin_state_up": true,
    "description": "simple pool",
    "lb_algorithm": "ROUND_ROBIN",
    "name": "my-pool",
    "protocol": "HTTP",
    "subnet_id": "e301aed0-d9e7-498a-977c-1bbfaf14ed5d"
  }
}
```

15.4.13.2. Response

Example 15.54. Create pool: JSON response

```
{
  "pool": {
    "status": "PENDING_CREATE",
    "lb_algorithm": "ROUND_ROBIN",
    "protocol": "HTTP",
    "description": "simple pool",
    "health_monitors": [],
    "members": [],
    "status_description": null,
    "id": "af95e0ce-8a26-4f29-9524-db41e7769c73",
    "vip_id": null,
    "name": "my-pool",
    "admin_state_up": true,
    "subnet_id": "e301aed0-d9e7-498a-977c-1bbfaf14ed5d",
    "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
    "health_monitors_status": [],
    "provider": "haproxy"
  }
}
```

15.4.14. Show pool details

Method	URI	Description
GET	/v2.0/lbaas/pools/{pool_id}	Shows details for a pool.

This operation shows details for a pool, by ID. If you are not an administrative user and the pool object does not belong to your tenant account, the call returns the HTTP Forbidden (403) response code.

If this operation succeeds, it returns a `pool` element.

Example: Show pool details

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.14.1. Request

This operation does not accept a request body.

15.4.14.2. Response

Example 15.55. Show pool details: JSON response

```
{
  "pool": {
    "status": "ACTIVE",
    "lb_algorithm": "ROUND_ROBIN",
    "protocol": "HTTP",
    "description": "",
    "health_monitors": [
      "b7633ade-24dc-4d72-8475-06aa22be5412"
    ],
    "members": [
      "cf024846-7516-4e3a-b0fb-6590322c836f"
    ],
    "status_description": null,
    "id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332",
    "vip_id": "388c739a-6a57-4e74-bc7b-a5cd60248bba",
    "name": "pool1",
    "admin_state_up": true,
    "subnet_id": "aa547115-d710-4d6d-bb2c-b038d9c2704b",
    "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
    "health_monitors_status": [
      {
        "monitor_id": "b7633ade-24dc-4d72-8475-06aa22be5412",
        "status": "ACTIVE",
        "status_description": null
      }
    ],
    "provider": "haproxy"
  }
}
```


15.4.15. Update pool

Method	URI	Description
PUT	/v2.0/lbaas/pools/{pool_id}	Updates a pool.

This operation updates the attributes of a pool. Upon successful validation of the request, the service returns the HTTP Accepted (202) response code.

Note: You cannot update the pool UUID, tenant_id, listener_id, listeners, health_monitor_id, protocol, and members immutable attributes. If you try to update any of these attributes, the service returns the HTTP Immutable (422) response code.

Note: You cannot update a pool if the load balancer to which it is attached does not have an ACTIVE provisioning status.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.15.1. Request

Example 15.56. Update pool: JSON request

```
{
  "pool": {
    "name": "SuperPool"
  }
}
```

15.4.15.2. Response

Example 15.57. Update pool: JSON response

```
{
  "pool": {
    "status": "PENDING_UPDATE",
    "lb_algorithm": "ROUND_ROBIN",
    "protocol": "HTTP",
    "description": "",
    "health_monitors": [
      "b7633ade-24dc-4d72-8475-06aa22be5412"
    ],
    "members": [
      "cf024846-7516-4e3a-b0fb-6590322c836f"
    ],
    "status_description": null,
    "id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332",
    "vip_id": "388c739a-6a57-4e74-bc7b-a5cd60248bba",
    "name": "SuperPool",
    "admin_state_up": true,
    "subnet_id": "aa547115-d710-4d6d-bb2c-b038d9c2704b",
    "tenant_id": "eabfef3fd1740a88a47ad98e132d238",
  }
}
```

```
    "health_monitors_status": [  
      {  
        "monitor_id": "b7633ade-24dc-4d72-8475-06aa22be5412",  
        "status": "ACTIVE",  
        "status_description": null  
      }  
    ],  
    "provider": "haproxy"  
  }  
}
```

15.4.16. Remove pool

Method	URI	Description
DELETE	/v2.0/lbaas/pools/{pool_id}	Removes a pool.

This operation removes a pool and its associated configuration from the tenant account. The API immediately purges any and all configuration data. You cannot recover it.

You cannot delete a pool if the load balancer to which it is attached does not have an ACTIVE provisioning status.

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.16.1. Request

This operation does not accept a request body.

15.4.17. List pool members

Method	URI	Description
GET	/v2.0/lbaas/pools/{pool_id}/members	Lists members of a pool.

Lists all members that are associated with a pool that is associated with your tenant account. The list of members includes only members that belong to the pool object identified by `pool_id`.

The list might be empty.

Example: List pool members

Normal response codes: 200

Error response codes: unauthorized (401), serviceUnavailable (503), Internal-server-error (500)

15.4.17.1. Request

This operation does not accept a request body.

15.4.17.2. Response

Example 15.58. List pool members: JSON response

```
{
  "members": [
    {
      "address": "10.0.0.8",
      "admin_state_up": true,
      "id": "9a7aff27-fd41-4ec1-ba4c-3eb92c629313",
      "protocol_port": 80,
      "subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
      "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
      "weight": 1
    }
  ]
}
```

15.4.18. Add member to pool

Method	URI	Description
POST	/v2.0/lbaas/pools/{pool_id}/members	Adds a member to a pool.

This operation provisions a member and adds it to a pool by using the configuration that you define in the request object. After the API validates the request and starts the provisioning process, it returns a response object, which contains a unique ID.

At a minimum, you must specify these pool attributes:

- `tenant_id`. Admin only. Required to create a pool for another tenant.
- `address`. The IP address of the member to receive traffic from the load balancer.
- `protocol_port`. The port on which the member listens for traffic.

Some attributes receive default values if you omit them from the request:

- `admin_state_up`. Default is `true`.
- `weight`. Default is `1`.

If you omit the `subnet_id` parameter, LBaaS uses the `vip_subnet_id` parameter value for the subnet UUID.

If the request fails due to incorrect data, the service returns the HTTP Bad Request (400) response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.

To configure all documented member features at creation time, specify additional elements or attributes in the request.

Administrative users can specify a tenant ID that is different than their own to create members for other tenants.

To update a member, the load balancer must have an ACTIVE provisioning status.

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.18.1. Request

Example 15.59. Add member to a pool: JSON request

```
{
  "member": {
    "address": "10.0.0.22",
    "admin_state_up": true,
    "protocol_port": "90",
```

```
    "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332",  
    "weight": "1"  
  }  
}
```

15.4.18.2. Response

Example 15.60. Add member to pool: JSON response

```
{  
  "member": {  
    "admin_state_up": true,  
    "status": "PENDING_CREATE",  
    "status_description": null,  
    "weight": 1,  
    "address": "10.0.1.22",  
    "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",  
    "protocol_port": 90,  
    "id": "cf024846-7516-4e3a-b0fb-6590322c836f",  
    "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"  
  }  
}
```

15.4.19. Show pool member details

Method	URI	Description
GET	/v2.0/lbaas/pools/{pool_id}/members/{member_id}	Shows details for a pool member.

This operation returns a member object identified by `member_id` that belongs to a pool object identified by `pool_id`. If you are not an administrative user and the pool or member object does not belong to your tenant account, the service returns the HTTP Forbidden (403) response code.

If this operation succeeds, it returns a pool element.

Example: Show pool member details

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.19.1. Request

This operation does not accept a request body.

15.4.19.2. Response

Example 15.61. Show pool member details: JSON response

```
{
  "member": {
    "admin_state_up": true,
    "status": "ACTIVE",
    "status_description": null,
    "weight": 1,
    "address": "10.0.1.22",
    "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
    "protocol_port": 90,
    "id": "cf024846-7516-4e3a-b0fb-6590322c836f",
    "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
  }
}
```

15.4.20. Update pool member

Method	URI	Description
PUT	/v2.0/lbaas/pools/{pool_id}/members/{member_id}	Updates attributes for a pool member.

Upon successful validation of the request, the service returns the HTTP OK (200) response code.

Note: You cannot update the member UUID, tenant_id, address, protocol_port, and subnet_id attributes. If you attempt to update any of these attributes, the service returns the HTTP Immutable (422) response code.

Note: You cannot update a member if the attached load balancer does not have an ACTIVE provisioning status.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.20.1. Request

Example 15.62. Update pool member: JSON request

```
{
  "member": {
    "weight": 5
  }
}
```

15.4.20.2. Response

Example 15.63. Update pool member: JSON response

```
{
  "member": {
    "admin_state_up": true,
    "status": "PENDING_UPDATE",
    "status_description": null,
    "weight": 5,
    "address": "10.0.1.22",
    "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
    "protocol_port": 90,
    "id": "cf024846-7516-4e3a-b0fb-6590322c836f",
    "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
  }
}
```

15.4.21. Remove member from pool

Method	URI	Description
DELETE	/v2.0/lbaas/pools/{pool_id}/members/{member_id}	Removes a member from a pool and its associated configuration from the tenant account.

The API immediately purges any and all configuration data. You cannot recover it.

You cannot delete a member if the attached load balancer does not have an ACTIVE provisioning status.

Example: Remove a member from a pool

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.21.1. Request

This operation does not accept a request body.

15.4.22. Create health monitor

Method	URI	Description
POST	/v2.0/lbaas/health_monitors	Creates a health monitor.

This operation provisions a health monitor by using the configuration that you define in the request object. After the API validates the request and start the provisioning process, it returns a response object. The object contains a unique identifier.

At a minimum, you must specify these health monitor attributes:

- `tenant_id`. Admin only. Required to create a health monitor for another tenant.
- `type`. The type of health monitor. A valid value is TCP, HTTP, or HTTPS.
- `delay`. The interval, in seconds, between health checks.
- `timeout`. The time, in seconds, after which a health check times out.
- `max_retries`. Number of failed health checks before marked as OFFLINE.
- `pool_id`. The pool to monitor.

Some attributes receive default values if you omit them from the request, and are only useful when you specify a health monitor type of HTTP(S):

- `http_method`. Default is GET.
- `url_path`. Default is /.
- `expected_codes`. The expected HTTP status codes to get from a successful health check. Default is 200.
- `admin_state_up`. Default is true.

If the API cannot fulfill the request due to insufficient data or data that is not valid, it returns the Bad Request (400) response code with information about the nature of the failure in the response body. Failures in the validation process are non-recoverable and require that you correct the cause of the failure and submit the request again.

You can configure all documented features of the health monitor at creation time by specifying the additional elements or attributes in the request.

Administrative users can specify a tenant ID that is different than their own to create health monitors for other tenants.

To update a health monitor, the load balancer to which to attach must have an ACTIVE provisioning status.

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.22.1. Request

Example 15.64. Create health monitor: JSON request

```
{
  "health_monitor": {
    "admin_state_up": true,
    "delay": "1",
    "expected_codes": "200,201,202",
    "http_method": "GET",
    "max_retries": 5,
    "timeout": 1,
    "type": "HTTP",
    "url_path": "/index.html"
  }
}
```

15.4.22.2. Response

Example 15.65. Create health monitor: JSON response

```
{
  "health_monitor": {
    "admin_state_up": true,
    "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
    "delay": 1,
    "expected_codes": "200,201,202",
    "max_retries": 5,
    "http_method": "GET",
    "timeout": 1,
    "pools": [],
    "url_path": "/index.html",
    "type": "HTTP",
    "id": "b7633ade-24dc-4d72-8475-06aa22be5412"
  }
}
```


15.4.23. List health monitors

Method	URI	Description
GET	/v2.0/lbaas/health_monitors	Lists health monitors.

This operation lists all health monitors that are associated with your tenant account.

This operation returns a list, which might be empty.

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

15.4.23.1. Request

This operation does not accept a request body.

15.4.23.2. Response

Example 15.66. List health monitors: JSON response

```
{
  "health_monitors": [
    {
      "admin_state_up": true,
      "tenant_id": "eabfefaf3fd1740a88a47ad98e132d238",
      "delay": 1,
      "expected_codes": "200,201,202",
      "max_retries": 5,
      "http_method": "GET",
      "timeout": 1,
      "pools": [
        {
          "status": "ACTIVE",
          "status_description": null,
          "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
        }
      ],
      "url_path": "/index.html",
      "type": "HTTP",
      "id": "b7633ade-24dc-4d72-8475-06aa22be5412"
    }
  ]
}
```

15.4.24. Show health monitor details

Method	URI	Description
GET	/v2.0/lbaas/health_monitors/{health_monitor_id}	Shows details for a health monitor.

This operation returns a health monitor object, by health monitor ID. If you are not an administrative user and the health monitor object does not belong to your tenant account, the service returns the HTTP Forbidden (403) response code.

Example: Show health monitor details

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.24.1. Request

This operation does not accept a request body.

15.4.24.2. Response

Example 15.67. Show health monitor details: JSON response

```
{
  "health_monitor": {
    "admin_state_up": true,
    "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
    "delay": 1,
    "expected_codes": "200,201,202",
    "max_retries": 5,
    "http_method": "GET",
    "timeout": 1,
    "pools": [
      {
        "status": "ACTIVE",
        "status_description": null,
        "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
      }
    ],
    "url_path": "/index.html",
    "type": "HTTP",
    "id": "b7633ade-24dc-4d72-8475-06aa22be5412"
  }
}
```

15.4.25. Update health monitor

Method	URI	Description
PUT	/v2.0/lbaas/health_monitors/{health_monitor_id}	Updates a health monitor.

Upon successful validation of the request, the service returns the HTTP Accepted (202) response code.

Note: The health monitor UUID, tenant_id, pool_id, and type are immutable attributes and cannot be updated. If you specify an unsupported attribute, the service returns the HTTP Immutable (422) response code.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.25.1. Request

Example 15.68. Update health monitor: JSON request

```
{
  "health_monitor": {
    "admin_state_up": false,
    "delay": "2",
    "expected_codes": "200",
    "http_method": "POST",
    "max_retries": 2,
    "timeout": 2,
    "url_path": "/page.html"
  }
}
```

15.4.25.2. Response

Example 15.69. Update health monitor: JSON response

```
{
  "health_monitor": {
    "admin_state_up": false,
    "tenant_id": "eabfefaf3fd1740a88a47ad98e132d238",
    "delay": 2,
    "expected_codes": "200",
    "max_retries": 2,
    "http_method": "POST",
    "timeout": 2,
    "pools": [
      {
        "status": "ACTIVE",
        "status_description": null,
        "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
      }
    ],
    "url_path": "/page.html",
  }
}
```

```
    "type": "HTTP",  
    "id": "b7633ade-24dc-4d72-8475-06aa22be5412"  
  }  
}
```

15.4.26. Remove health monitor

Method	URI	Description
DELETE	/v2.0/lbaas/health_monitors/{health_monitor_id}	Removes a health monitor and its associated configuration from the tenant account.

The API immediately purges any and all configuration data. You cannot recover it.

You cannot delete a health monitor if the attached load balancer does not have an ACTIVE provisioning status.

Example: Delete a health monitor

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.4.26.1. Request

This operation does not accept a request body.

15.5. LBaaS 1.0 (DEPRECATED) (lb, vips, health_monitors, pools, members)

The Load-Balancer-as-a-Service (LBaaS) v1.0 extension pairs with the Networking v2.0 API to enable OpenStack tenants to manage load balancers for their VMs. With this extension, you can load-balance client traffic from one network to application services, such as VMs, on the same network.

Use this extension to create and manage virtual IP addresses (VIPs), pools, members of a pool, health monitors, and view status of a resource.

Table 15.2. Load balancer statuses

Status	Description
ACTIVE	The resource is ready and active.
PENDING_CREATE	The resource is being created.
PENDING_UPDATE	The resource is being updated.
PENDING_DELETE	The resource is pending deletion.
INACTIVE	The resource is not active.
ERROR	An object within the service is not working. The <code>error_details</code> attribute provides an explanation for the error, its cause, and possibly a solution.

Method	URI	Description
GET	/v2.0/lb/vips	Lists VIPs.
POST	/v2.0/lb/vips	Creates a load balancer VIP.
GET	/v2.0/lb/vips/{vip_id}	Shows details for a VIP.

Method	URI	Description
PUT	/v2.0/lb/vips/{vip_id}	Updates a load balancer VIP.
DELETE	/v2.0/lb/vips/{vip_id}	Deletes a load balancer VIP.
GET	/v2.0/lb/health_monitors	Lists health monitors.
POST	/v2.0/lb/health_monitors	Creates a load balancer health monitor.
GET	/v2.0/lb/health_monitors/{health_monitor_id}	Shows details for a health monitor.
PUT	/v2.0/lb/health_monitors/{health_monitor_id}	Updates a load balancer health monitor.
DELETE	/v2.0/lb/health_monitors/{health_monitor_id}	Deletes a load balancer health monitor.
GET	/v2.0/lb/pools	Lists pools.
POST	/v2.0/lb/pools	Creates a load balancer pool.
GET	/v2.0/lb/pools/{pool_id}	Shows details for a pool.
PUT	/v2.0/lb/pools/{pool_id}	Updates a load balancer pool.
DELETE	/v2.0/lb/pools/{pool_id}	Deletes a load balancer pool.
POST	/v2.0/lb/pools/{pool_id}/health_monitors	Associates a health monitor with a pool.
DELETE	/v2.0/lb/pools/{pool_id}/health_monitors/{health_monitor_id}	Disassociates a health monitor from a pool.
GET	/v2.0/lb/members	Lists members.
POST	/v2.0/lb/members	Creates a load balancer member.
GET	/v2.0/lb/members/{member_id}	Shows details for a member.
PUT	/v2.0/lb/members/{member_id}	Updates a load balancer member.
DELETE	/v2.0/lb/members/{member_id}	Deletes a load balancer member.

15.5.1. List VIPs

Method	URI	Description
GET	/v2.0/lb/vips	Lists VIPs.

The list might be empty.

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

15.5.1.1. Request

This operation does not accept a request body.

15.5.1.2. Response

Example 15.70. List VIPs: JSON response

```
{
  "vips": [
    {
      "status": "ACTIVE",
      "protocol": "HTTP",
      "description": "",
      "address": "10.0.0.4",
      "protocol_port": 80,
      "port_id": "5328aeaa-2988-41c0-b5fe-0fd0660979d3",
      "id": "388c739a-6a57-4e74-bc7b-a5cd60248bba",
      "status_description": null,
      "name": "my-Vip",
      "admin_state_up": true,
      "subnet_id": "aa547115-d710-4d6d-bb2c-b038d9c2704b",
      "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
      "connection_limit": -1,
      "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332",
      "session_persistence": null
    }
  ]
}
```

15.5.2. Create a load balancer VIP

Method	URI	Description
POST	/v2.0/lb/vips	Creates a load balancer VIP.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.2.1. Request

Example 15.71. Create a load balancer VIP: JSON request

```
{
  "vip": {
    "protocol": "HTTP",
    "name": "NewVip",
    "admin_state_up": true,
    "subnet_id": "0ba2ef27-0054-4b28-a8fa-f215e8079272",
    "pool_id": "105320c3-8416-4997-9c1c-4098b95fdaca",
    "protocol_port": "80"
  }
}
```

15.5.2.2. Response

Example 15.72. Create a load balancer VIP: JSON response

```
{
  "vip": {
    "status": "PENDING_CREATE",
    "protocol": "HTTP",
    "description": "",
    "address": "10.0.0.4",
    "protocol_port": 80,
    "port_id": "0ba4cd9c-edb4-4594-bac4-b68b49d5f04c",
    "id": "fa0373e0-9dd4-4ff7-98fc-8cceca9bdb4e",
    "status_description": null,
    "name": "NewVip",
    "admin_state_up": true,
    "subnet_id": "0ba2ef27-0054-4b28-a8fa-f215e8079272",
    "tenant_id": "e68c3e65e1f34ee9b2357d0fe418a78b",
    "connection_limit": -1,
    "pool_id": "105320c3-8416-4997-9c1c-4098b95fdaca",
    "session_persistence": null
  }
}
```


15.5.3. Show VIP details

Method	URI	Description
GET	/v2.0/lb/vips/{vip_id}	Shows details for a VIP.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.3.1. Request

This table shows the URI parameters for the show vip details request:

Name	Type	Description
{vip_id}	UUID	The UUID for the VIP.

This operation does not accept a request body.

15.5.3.2. Response

Example 15.73. Show VIP details: JSON response

```
{
  "vip": {
    "status": "ACTIVE",
    "protocol": "HTTP",
    "description": "",
    "address": "10.0.0.4",
    "protocol_port": 80,
    "port_id": "5328aeea-2988-41c0-b5fe-0fd0660979d3",
    "id": "388c739a-6a57-4e74-bc7b-a5cd60248bba",
    "status_description": null,
    "name": "my-Vip",
    "admin_state_up": true,
    "subnet_id": "aa547115-d710-4d6d-bb2c-b038d9c2704b",
    "tenant_id": "eabfefaf3fd1740a88a47ad98e132d238",
    "connection_limit": -1,
    "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332",
    "session_persistence": null
  }
}
```

15.5.4. Update VIP

Method	URI	Description
PUT	/v2.0/lb/vips/{vip_id}	Updates a load balancer VIP.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.4.1. Request

This table shows the URI parameters for the update vip request:

Name	Type	Description
{vip_id}	UUID	The UUID for the VIP.

Example 15.74. Update VIP: JSON request

```
{
  "vip": {
    "connection_limit": "1000"
  }
}
```

15.5.4.2. Response

Example 15.75. Update VIP: JSON response

```
{
  "vip": {
    "status": "PENDING_UPDATE",
    "protocol": "HTTP",
    "description": "",
    "address": "10.0.0.4",
    "protocol_port": 80,
    "port_id": "0ba4cd9c-edb4-4594-bac4-b68b49d5f04c",
    "id": "fa0373e0-9dd4-4ff7-98fc-8cceca9bdb4e",
    "status_description": null,
    "name": "NewVip",
    "admin_state_up": true,
    "subnet_id": "0ba2ef27-0054-4b28-a8fa-f215e8079272",
    "tenant_id": "e68c3e65e1f34ee9b2357d0fe418a78b",
    "connection_limit": 1000,
    "pool_id": "105320c3-8416-4997-9c1c-4098b95fdaca",
    "session_persistence": null
  }
}
```

15.5.5. Delete VIP

Method	URI	Description
DELETE	/v2.0/lb/vips/{vip_id}	Deletes a load balancer VIP.

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.5.1. Request

This table shows the URI parameters for the delete vip request:

Name	Type	Description
{vip_id}	UUID	The UUID for the VIP.

This operation does not accept a request body.

15.5.6. List health monitors

Method	URI	Description
GET	/v2.0/lb/health_monitors	Lists health monitors.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.6.1. Request

This operation does not accept a request body.

15.5.6.2. Response

Example 15.76. List health monitors: JSON response

```
{
  "health_monitors": [
    {
      "admin_state_up": true,
      "tenant_id": "eabfef3fd1740a88a47ad98e132d238",
      "delay": 1,
      "expected_codes": "200,201,202",
      "max_retries": 5,
      "http_method": "GET",
      "timeout": 1,
      "pools": [
        {
          "status": "ACTIVE",
          "status_description": null,
          "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
        }
      ],
      "url_path": "/index.html",
      "type": "HTTP",
      "id": "b7633ade-24dc-4d72-8475-06aa22be5412"
    }
  ]
}
```

15.5.7. Create a load balancer health monitor

Method	URI	Description
POST	/v2.0/lb/health_monitors	Creates a load balancer health monitor.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.7.1. Request

Example 15.77. Create a load balancer health monitor: JSON request

```
{
  "health_monitor": {
    "admin_state_up": true,
    "delay": "1",
    "expected_codes": "200,201,202",
    "http_method": "GET",
    "max_retries": 5,
    "timeout": 1,
    "type": "HTTP",
    "url_path": "/index.html"
  }
}
```

15.5.7.2. Response

Example 15.78. Create a load balancer health monitor: JSON response

```
{
  "health_monitor": {
    "admin_state_up": true,
    "tenant_id": "eabfef3fd1740a88a47ad98e132d238",
    "delay": 1,
    "expected_codes": "200,201,202",
    "max_retries": 5,
    "http_method": "GET",
    "timeout": 1,
    "pools": [],
    "url_path": "/index.html",
    "type": "HTTP",
    "id": "b7633ade-24dc-4d72-8475-06aa22be5412"
  }
}
```

15.5.8. Show health monitor details

Method	URI	Description
GET	/v2.0/lb/health_monitors/{health_monitor_id}	Shows details for a health monitor.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.8.1. Request

This table shows the URI parameters for the show health monitor details request:

Name	Type	Description
{health_monitor_id}	UUID	The UUID for the health monitor.

This operation does not accept a request body.

15.5.8.2. Response

Example 15.79. Show health monitor details: JSON response

```
{
  "health_monitor": {
    "admin_state_up": true,
    "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
    "delay": 1,
    "expected_codes": "200,201,202",
    "max_retries": 5,
    "http_method": "GET",
    "timeout": 1,
    "pools": [
      {
        "status": "ACTIVE",
        "status_description": null,
        "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
      }
    ],
    "url_path": "/index.html",
    "type": "HTTP",
    "id": "b7633ade-24dc-4d72-8475-06aa22be5412"
  }
}
```

15.5.9. Update health monitor

Method	URI	Description
PUT	/v2.0/lb/health_monitors/{health_monitor_id}	Updates a load balancer health monitor.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.9.1. Request

This table shows the URI parameters for the update health monitor request:

Name	Type	Description
{health_monitor_id}	UUID	The UUID for the health monitor.

Example 15.80. Update health monitor: JSON request

```
{
  "health_monitor": {
    "admin_state_up": false,
    "delay": "2",
    "expected_codes": "200",
    "http_method": "POST",
    "max_retries": 2,
    "timeout": 2,
    "url_path": "/page.html"
  }
}
```

15.5.9.2. Response

Example 15.81. Update health monitor: JSON response

```
{
  "health_monitor": {
    "admin_state_up": false,
    "tenant_id": "eabfefaf3fd1740a88a47ad98e132d238",
    "delay": 2,
    "expected_codes": "200",
    "max_retries": 2,
    "http_method": "POST",
    "timeout": 2,
    "pools": [
      {
        "status": "ACTIVE",
        "status_description": null,
        "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
      }
    ],
    "url_path": "/page.html",
    "type": "HTTP",
    "id": "b7633ade-24dc-4d72-8475-06aa22be5412"
  }
}
```

```
}  
}
```


15.5.10. Delete health monitor

Method	URI	Description
DELETE	/v2.0/lb/health_monitors/{health_monitor_id}	Deletes a load balancer health monitor.

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.10.1. Request

This table shows the URI parameters for the delete health monitor request:

Name	Type	Description
{health_monitor_id}	UUID	The UUID for the health monitor.

15.5.11. List pools

Method	URI	Description
GET	/v2.0/lb/pools	Lists pools.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.11.1. Request

This operation does not accept a request body.

15.5.11.2. Response

Example 15.82. List pools: JSON response

```
{
  "pools": [
    {
      "status": "ACTIVE",
      "lb_algorithm": "ROUND_ROBIN",
      "protocol": "HTTP",
      "description": "",
      "health_monitors": [
        "b7633ade-24dc-4d72-8475-06aa22be5412"
      ],
      "members": [
        "cf024846-7516-4e3a-b0fb-6590322c836f"
      ],
      "status_description": null,
      "id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332",
      "vip_id": "388c739a-6a57-4e74-bc7b-a5cd60248bba",
      "name": "pool1",
      "admin_state_up": true,
      "subnet_id": "aa547115-d710-4d6d-bb2c-b038d9c2704b",
      "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
      "health_monitors_status": [
        {
          "monitor_id": "b7633ade-24dc-4d72-8475-06aa22be5412",
          "status": "ACTIVE",
          "status_description": null
        }
      ],
      "provider": "haproxy"
    }
  ]
}
```

15.5.12. Create a load balancer pool

Method	URI	Description
POST	/v2.0/lb/pools	Creates a load balancer pool.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.12.1. Request

Example 15.83. Create a load balancer pool: JSON request

```
{
  "pool": {
    "admin_state_up": true,
    "description": "simple pool",
    "lb_algorithm": "ROUND_ROBIN",
    "name": "my-pool",
    "protocol": "HTTP",
    "subnet_id": "e301aed0-d9e7-498a-977c-1bbfaf14ed5d"
  }
}
```

15.5.12.2. Response

Example 15.84. Create a load balancer pool: JSON response

```
{
  "pool": {
    "status": "PENDING_CREATE",
    "lb_algorithm": "ROUND_ROBIN",
    "protocol": "HTTP",
    "description": "simple pool",
    "health_monitors": [],
    "members": [],
    "status_description": null,
    "id": "af95e0ce-8a26-4f29-9524-db41e7769c73",
    "vip_id": null,
    "name": "my-pool",
    "admin_state_up": true,
    "subnet_id": "e301aed0-d9e7-498a-977c-1bbfaf14ed5d",
    "tenant_id": "eabfefaf3fd1740a88a47ad98e132d238",
    "health_monitors_status": [],
    "provider": "haproxy"
  }
}
```

15.5.13. Show pool details

Method	URI	Description
GET	/v2.0/lb/pools/{pool_id}	Shows details for a pool.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.13.1. Request

This table shows the URI parameters for the show pool details request:

Name	Type	Description
{pool_id}	UUID	The UUID for the pool.

This operation does not accept a request body.

15.5.13.2. Response

Example 15.85. Show pool details: JSON response

```
{
  "pool": {
    "status": "ACTIVE",
    "lb_algorithm": "ROUND_ROBIN",
    "protocol": "HTTP",
    "description": "",
    "health_monitors": [
      "b7633ade-24dc-4d72-8475-06aa22be5412"
    ],
    "members": [
      "cf024846-7516-4e3a-b0fb-6590322c836f"
    ],
    "status_description": null,
    "id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332",
    "vip_id": "388c739a-6a57-4e74-bc7b-a5cd60248bba",
    "name": "pool1",
    "admin_state_up": true,
    "subnet_id": "aa547115-d710-4d6d-bb2c-b038d9c2704b",
    "tenant_id": "eabfefaf3fd1740a88a47ad98e132d238",
    "health_monitors_status": [
      {
        "monitor_id": "b7633ade-24dc-4d72-8475-06aa22be5412",
        "status": "ACTIVE",
        "status_description": null
      }
    ],
    "provider": "haproxy"
  }
}
```

15.5.14. Update pool

Method	URI	Description
PUT	/v2.0/lb/pools/{pool_id}	Updates a load balancer pool.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.14.1. Request

This table shows the URI parameters for the update pool request:

Name	Type	Description
{pool_id}	UUID	The UUID for the pool.

Example 15.86. Update pool: JSON request

```
{
  "pool": {
    "name": "SuperPool"
  }
}
```

15.5.14.2. Response

Example 15.87. Update pool: JSON response

```
{
  "pool": {
    "status": "PENDING_UPDATE",
    "lb_algorithm": "ROUND_ROBIN",
    "protocol": "HTTP",
    "description": "",
    "health_monitors": [
      "b7633ade-24dc-4d72-8475-06aa22be5412"
    ],
    "members": [
      "cf024846-7516-4e3a-b0fb-6590322c836f"
    ],
    "status_description": null,
    "id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332",
    "vip_id": "388c739a-6a57-4e74-bc7b-a5cd60248bba",
    "name": "SuperPool",
    "admin_state_up": true,
    "subnet_id": "aa547115-d710-4d6d-bb2c-b038d9c2704b",
    "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
    "health_monitors_status": [
      {
        "monitor_id": "b7633ade-24dc-4d72-8475-06aa22be5412",
        "status": "ACTIVE",
        "status_description": null
      }
    ]
  }
}
```

```
    "provider": "haproxy"  
  }  
}
```

15.5.15. Delete pool

Method	URI	Description
DELETE	/v2.0/lb/pools/{pool_id}	Deletes a load balancer pool.

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.15.1. Request

This table shows the URI parameters for the delete pool request:

Name	Type	Description
{pool_id}	UUID	The UUID for the pool.

This operation does not accept a request body.

15.5.16. Associate health monitor with pool

Method	URI	Description
POST	/v2.0/lb/pools/{pool_id}/health_monitors	Associates a health monitor with a pool.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.16.1. Request

This table shows the URI parameters for the associate health monitor with pool request:

Name	Type	Description
{pool_id}	UUID	The UUID for the pool.

Example 15.88. Associate health monitor with pool: JSON request

```
{
  "health_monitor": {
    "id": "b624decf-d5d3-4c66-9a3d-f047e7786181"
  }
}
```

15.5.16.2. Response

Example 15.89. Associate health monitor with pool: JSON response

```
{
  "health_monitor": {}
}
```


15.5.17. Disassociate health monitor from pool

Method	URI	Description
DELETE	/v2.0/lb/pools/{pool_id}/health_monitors/{health_monitor_id}	Disassociates a health monitor from a pool.

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.17.1. Request

This table shows the URI parameters for the disassociate health monitor from pool request:

Name	Type	Description
{pool_id}	UUID	The UUID for the pool.
{health_monitor_id}	UUID	The UUID for the health monitor.

This operation does not accept a request body.

15.5.18. List members

Method	URI	Description
GET	/v2.0/lb/members	Lists members.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.18.1. Request

This operation does not accept a request body.

15.5.18.2. Response

Example 15.90. List members: JSON response

```
{
  "members": [
    {
      "admin_state_up": true,
      "status": "ACTIVE",
      "status_description": null,
      "weight": 1,
      "address": "10.0.1.22",
      "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
      "protocol_port": 90,
      "id": "cf024846-7516-4e3a-b0fb-6590322c836f",
      "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
    }
  ]
}
```

15.5.19. Create a load balancer member

Method	URI	Description
POST	/v2.0/lb/members	Creates a load balancer member.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.19.1. Request

Example 15.91. Create a load balancer member: JSON request

```
{
  "member": {
    "address": "10.0.0.22",
    "admin_state_up": true,
    "protocol_port": "90",
    "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332",
    "weight": "1"
  }
}
```

15.5.19.2. Response

Example 15.92. Create a load balancer member: JSON response

```
{
  "member": {
    "admin_state_up": true,
    "status": "PENDING_CREATE",
    "status_description": null,
    "weight": 1,
    "address": "10.0.1.22",
    "tenant_id": "eabfef3fd1740a88a47ad98e132d238",
    "protocol_port": 90,
    "id": "cf024846-7516-4e3a-b0fb-6590322c836f",
    "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
  }
}
```

15.5.20. Show member details

Method	URI	Description
GET	/v2.0/lb/members/{member_id}	Shows details for a member.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.20.1. Request

This table shows the URI parameters for the show member details request:

Name	Type	Description
{member_id}	UUID	The UUID for the member.

This operation does not accept a request body.

15.5.20.2. Response

Example 15.93. Show member details: JSON response

```
{
  "member": {
    "admin_state_up": true,
    "status": "ACTIVE",
    "status_description": null,
    "weight": 1,
    "address": "10.0.1.22",
    "tenant_id": "eabfefa3fd1740a88a47ad98e132d238",
    "protocol_port": 90,
    "id": "cf024846-7516-4e3a-b0fb-6590322c836f",
    "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
  }
}
```

15.5.21. Update member

Method	URI	Description
PUT	/v2.0/lb/members/{member_id}	Updates a load balancer member.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.21.1. Request

This table shows the URI parameters for the update member request:

Name	Type	Description
{member_id}	UUID	The UUID for the member.

Example 15.94. Update member: JSON request

```
{
  "member": {
    "weight": 5
  }
}
```

15.5.21.2. Response

Example 15.95. Update member: JSON response

```
{
  "member": {
    "admin_state_up": true,
    "status": "PENDING_UPDATE",
    "status_description": null,
    "weight": 5,
    "address": "10.0.1.22",
    "tenant_id": "eabfefaf3fd1740a88a47ad98e132d238",
    "protocol_port": 90,
    "id": "cf024846-7516-4e3a-b0fb-6590322c836f",
    "pool_id": "5a9a3e9e-d1aa-448e-af37-a70171f2a332"
  }
}
```

15.5.22. Delete member

Method	URI	Description
DELETE	/v2.0/lb/members/{member_id}	Deletes a load balancer member.

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.5.22.1. Request

This table shows the URI parameters for the delete member request:

Name	Type	Description
{member_id}	UUID	The UUID for the member.

This operation does not accept a request body.

15.6. Metering labels and rules (metering-labels, metering-label-rules)

Creates, modifies, and deletes OpenStack Layer3 metering labels and rules.

Method	URI	Description
GET	/v2.0/metering/metering-labels	Lists all L3 metering labels that belong to the tenant.
POST	/v2.0/metering/metering-labels	Creates an L3 metering label.
GET	/v2.0/metering/metering-labels/{metering-label-id}	Shows details for a metering label.
DELETE	/v2.0/metering/metering-labels/{metering-label-id}	Deletes an L3 metering label.
GET	/v2.0/metering/metering-label-rules	Lists a summary of all L3 metering label rules that belong to the tenant.
POST	/v2.0/metering/metering-label-rules	Creates an L3 metering label rule.
GET	/v2.0/metering/metering-label-rules/{metering-label-rule-id}	Shows details for a metering label rule.
DELETE	/v2.0/metering/metering-label-rules/{metering-label-rule-id}	Deletes an L3 metering label rule.

15.6.1. List metering labels

Method	URI	Description
GET	/v2.0/metering/metering-labels	Lists all L3 metering labels that belong to the tenant.

The list shows the UUID for each metering label.

Normal response codes: 200

Error response codes: unauthorized (401)

15.6.1.1. Request

Example 15.96. List metering labels: JSON request

```
GET /v2.0/metering/metering-labels HTTP/1.1
Host: controlnode:9696
User-Agent: python-neutronclient
Content-Type: application/json
Accept: application/json
X-Auth-Token: c52a1b304fec4ca0ac85dc1741eec6e2
```

This operation does not accept a request body.

15.6.1.2. Response

Example 15.97. List metering labels: JSON response

```
{
  "metering_labels": [
    {
      "tenant_id": "45345b0ee1ea477fac0f541b2cb79cd4",
      "description": "label1 description",
      "name": "label1",
      "id": "a6700594-5b7a-4105-8bfe-723b346ce866",
      "shared": false
    },
    {
      "tenant_id": "45345b0ee1ea477fac0f541b2cb79cd4",
      "description": "label2 description",
      "name": "label2",
      "id": "e131d186-b02d-4c0b-83d5-0c0725c4f812",
      "shared": false
    }
  ]
}
```

15.6.2. Create metering label

Method	URI	Description
POST	/v2.0/metering/metering-labels	Creates an L3 metering label.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), forbidden (403)

15.6.2.1. Request

Example 15.98. Create metering label: JSON request

```
{
  "metering_label": {
    "name": "label1",
    "description": "description of label1"
  }
}
```

15.6.2.2. Response

Example 15.99. Create metering label: JSON response

```
{
  "metering_label": {
    "tenant_id": "45345b0ee1ea477fac0f541b2cb79cd4",
    "description": "description of label1",
    "name": "label1",
    "id": "bc91b832-8465-40a7-a5d8-ba87de442266",
    "shared": false
  }
}
```


15.6.3. Show metering label details

Method	URI	Description
GET	/v2.0/metering/metering-labels/{metering-label-id}	Shows details for a metering label.

The response body shows the description, name, and UUID.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

15.6.3.1. Request

This table shows the URI parameters for the show metering label details request:

Name	Type	Description
{metering_label_id}	UUID	The UUID of the metering label.

Example 15.100. Show metering label details: JSON request

```
GET /v2.0/metering/metering-labels/a6700594-5b7a-4105-8bfe-723b346ce866 HTTP/1.1
Host: controlnode:9696
User-Agent: python-neutronclient
Content-Type: application/json
Accept: application/json
X-Auth-Token: c52a1b304fec4ca0ac85dc1741eec6e2
```

This operation does not accept a request body.

15.6.3.2. Response

Example 15.101. Show metering label details: JSON response

```
{
  "metering_label": {
    "tenant_id": "45345b0ee1ea477fac0f541b2cb79cd4",
    "description": "label1 description",
    "name": "label1",
    "id": "a6700594-5b7a-4105-8bfe-723b346ce866",
    "shared": false
  }
}
```

15.6.4. Delete metering label

Method	URI	Description
DELETE	/v2.0/metering/metering-labels/{metering-label-id}	Deletes an L3 metering label.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404)

15.6.4.1. Request

This table shows the URI parameters for the delete metering label request:

Name	Type	Description
{metering_label_id}	UUID	The UUID of the metering label.

Example 15.102. Delete metering label: JSON request

```
DELETE /v2.0/metering/metering-labels/a6700594-5b7a-4105-8bfe-723b346ce866
HTTP/1.1
Host: controlnode:9696
User-Agent: python-neutronclient
Content-Type: application/json
Accept: application/json
X-Auth-Token: c52a1b304fec4ca0ac85dc1741eec6e2
```

This operation does not accept a request body.

15.6.4.2. Response

Example 15.103. Delete metering label: JSON response

```
status: 204
```

This operation does not return a response body.

15.6.5. List metering label rules

Method	URI	Description
GET	/v2.0/metering/metering-label-rules	Lists a summary of all L3 metering label rules that belong to the tenant.

The list shows the UUID for each metering label rule.

Normal response codes: 200

Error response codes: unauthorized (401)

15.6.5.1. Request

Example 15.104. List metering label rules: JSON request

```
GET /v2.0/metering/metering-label-rules HTTP/1.1
Host: controlnode:9696
User-Agent: python-neutronclient
Content-Type: application/json
Accept: application/json
X-Auth-Token: c52a1b304fec4ca0ac85dc1741eec6e2
```

This operation does not accept a request body.

15.6.5.2. Response

Example 15.105. List metering label rules: JSON response

```
{
  "metering_label_rules": [
    {
      "remote_ip_prefix": "20.0.0.0/24",
      "direction": "ingress",
      "metering_label_id": "e131d186-b02d-4c0b-83d5-0c0725c4f812",
      "id": "9536641a-7d14-4dc5-afaf-93a973ce0eb8",
      "excluded": false
    },
    {
      "remote_ip_prefix": "10.0.0.0/24",
      "direction": "ingress",
      "metering_label_id": "e131d186-b02d-4c0b-83d5-0c0725c4f812",
      "id": "ffc6fd15-40de-4e7d-b617-34d3f7a93aec",
      "excluded": false
    }
  ]
}
```

15.6.6. Create metering label rule

Method	URI	Description
POST	/v2.0/metering/metering-label-rules	Creates an L3 metering label rule.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), conflict (409)

15.6.6.1. Request

Example 15.106. Create metering label rule: JSON request

```
{
  "metering_label_rule": {
    "remote_ip_prefix": "10.0.1.0/24",
    "direction": "ingress",
    "metering_label_id": "e131d186-b02d-4c0b-83d5-0c0725c4f812"
  }
}
```

15.6.6.2. Response

Example 15.107. Create metering label rule: JSON response

```
{
  "metering_label_rule": {
    "remote_ip_prefix": "10.0.1.0/24",
    "direction": "ingress",
    "metering_label_id": "e131d186-b02d-4c0b-83d5-0c0725c4f812",
    "id": "00e13b58-b4f2-4579-9c9c-7ac94615f9ae",
    "excluded": false
  }
}
```

15.6.7. Show metering label rule details

Method	URI	Description
GET	/v2.0/metering/metering-label-rules/{metering-label-rule-id}	Shows details for a metering label rule.

The response body shows this information for each metering label rule:

- **direction.** Either ingress or egress.
- **excluded.** Either true or false.
- The UUID for the metering label rule.
- The remote IP prefix.
- The metering label ID for the metering label with which the rule is associated.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

15.6.7.1. Request

This table shows the URI parameters for the show metering label rule details request:

Name	Type	Description
{metering_label_rule_id}	UUID	The UUID of the metering label rule.

Example 15.108. Show metering label rule details: JSON request

```
GET /v2.0/metering/metering-label-rules/9536641a-7d14-4dc5-afaf-93a973ce0eb8
HTTP/1.1
Host: controlnode:9696
User-Agent: python-neutronclient
Content-Type: application/json
Accept: application/json
X-Auth-Token: c52a1b304fec4ca0ac85dc1741eec6e2
```

This operation does not accept a request body.

15.6.7.2. Response

Example 15.109. Show metering label rule details: JSON response

```
{
  "metering_label_rule": {
    "remote_ip_prefix": "20.0.0.0/24",
    "direction": "ingress",
    "metering_label_id": "e131d186-b02d-4c0b-83d5-0c0725c4f812",
    "id": "9536641a-7d14-4dc5-afaf-93a973ce0eb8",
    "excluded": false
  }
}
```


15.6.8. Delete metering label rule

Method	URI	Description
DELETE	/v2.0/metering/metering-label-rules/{metering-label-rule-id}	Deletes an L3 metering label rule.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404)

15.6.8.1. Request

This table shows the URI parameters for the delete metering label rule request:

Name	Type	Description
{metering_label_rule_id}	UUID	The UUID of the metering label rule.

Example 15.110. Delete metering label rule: JSON request

```
DELETE /v2.0/metering/metering-labels/37b31179-71ee-4f0a-b130-0eeb28e7ede7
HTTP/1.1
Host: controlnode:9696
User-Agent: python-neutronclient
Content-Type: application/json
Accept: application/json
X-Auth-Token: c52a1b304fec4ca0ac85dc1741eec6e2
```

This operation does not accept a request body.

15.6.8.2. Response

Example 15.111. Delete metering label rule: JSON response

```
status: 204
```

This operation does not return a response body.

15.7. Networks provider extended attributes (networks)

Lists, creates, shows information for, updates, and deletes networks.

The provider extension decorates network resources with additional attributes. These attributes are `provider:network_type`, `provider:physical_network`, and `provider:segmentation_id`. The validation rules for these attributes are the same as for the [Networks multiple provider extension](#). You cannot use both extensions at the same time.

Method	URI	Description
GET	/v2.0/networks	Lists networks that are accessible to the tenant who submits the request.

Method	URI	Description
POST	/v2.0/networks	Creates a network.
GET	/v2.0/networks/{network_id}	Shows details for a network.
PUT	/v2.0/networks/{network_id}	Updates a network.
DELETE	/v2.0/networks/{network_id}	Deletes a network.

15.7.1. List networks

Method	URI	Description
GET	/v2.0/networks	Lists networks that are accessible to the tenant who submits the request.

Normal response codes: 200

Error response codes: unauthorized (401)

15.7.1.1. Request

This operation does not accept a request body.

15.7.1.2. Response

Example 15.112. List networks: JSON response

```
{
  "networks": [
    {
      "status": "ACTIVE",
      "subnets": [
        "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
      ],
      "name": "private-network",
      "provider:physical_network": null,
      "admin_state_up": true,
      "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
      "provider:network_type": "local",
      "router:external": true,
      "mtu": 0,
      "shared": true,
      "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
      "provider:segmentation_id": null
    },
    {
      "status": "ACTIVE",
      "subnets": [
        "08eae331-0402-425a-923c-34f7cfe39c1b"
      ],
      "name": "private",
      "provider:physical_network": null,
      "admin_state_up": true,
      "tenant_id": "26a7980765d0414dbc1fc1f88cdb7e6e",
      "provider:network_type": "local",
      "router:external": true,
      "mtu": 0,
      "shared": true,
      "id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",
      "provider:segmentation_id": null
    }
  ]
}
```

15.7.2. Create network

Method	URI	Description
POST	/v2.0/networks	Creates a network.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

15.7.2.1. Request

Example 15.113. Create network: JSON request

```
{
  "network": {
    "name": "sample_network",
    "admin_state_up": true
  }
}
```

15.7.2.2. Response

Example 15.114. Create network: JSON response

```
{
  "network": {
    "status": "ACTIVE",
    "subnets": [],
    "name": "net1",
    "admin_state_up": true,
    "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
    "router:external": false,
    "mtu": 0,
    "shared": false,
    "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c"
  }
}
```

15.7.3. Show network details

Method	URI	Description
GET	/v2.0/networks/{network_id}	Shows details for a network.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

15.7.3.1. Request

This table shows the URI parameters for the show network details request:

Name	Type	Description
{network_id}	UUID	The UUID of the network.

This operation does not accept a request body.

15.7.3.2. Response

Example 15.115. Show network details: JSON response

```
{
  "network": {
    "status": "ACTIVE",
    "subnets": [
      "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
    ],
    "name": "private-network",
    "router:external": false,
    "admin_state_up": true,
    "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
    "created_at": "2016-03-08T20:19:41",
    "mtu": 0,
    "shared": true,
    "port_security_enabled": true,
    "updated_at": "2016-03-08T20:19:41",
    "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22"
  }
}
```

15.7.4. Update network

Method	URI	Description
PUT	/v2.0/networks/{network_id}	Updates a network.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.7.4.1. Request

This table shows the URI parameters for the update network request:

Name	Type	Description
{network_id}	UUID	The UUID of the network.

Example 15.116. Update network: JSON request

```
{
  "network": {
    "name": "sample_network_5_updated"
  }
}
```

15.7.4.2. Response

Example 15.117. Update network: JSON response

```
{
  "network": {
    "status": "ACTIVE",
    "subnets": [],
    "name": "sample_network_5_updated",
    "provider:physical_network": null,
    "admin_state_up": true,
    "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
    "provider:network_type": "local",
    "router:external": false,
    "mtu": 0,
    "shared": false,
    "port_security_enabled": true,
    "id": "1f370095-98f6-4079-be64-6d3d4a6adcc6",
    "provider:segmentation_id": null
  }
}
```

15.7.5. Delete network

Method	URI	Description
DELETE	/v2.0/networks/{network_id}	Deletes a network.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.7.5.1. Request

This table shows the URI parameters for the delete network request:

Name	Type	Description
{network_id}	UUID	The UUID of the network.

This operation does not accept a request body.

15.8. Networks multiple provider extension (networks)

Enables administrative users to define multiple physical bindings for an OpenStack Networking network and list or show details for networks with multiple physical bindings.

You cannot update any provider attributes. If you try to do so, an error occurs.

To delete a network with multiple physical bindings, issue a normal delete network request.

To define multiple physical bindings for a network, include a segments list in the request body of a POST /v2.0/networks request. Each element in the segments list has the same structure as the provider network attributes. These attributes are provider:network_type, provider:physical_network, and provider:segmentation_id. The validation rules for these attributes are the same as for the [Networks provider extended attributes](#). You cannot use both extensions at the same time.

The NSX and ML2 plug-ins support this extension. With the ML2 plug-in, you can specify multiple VLANs for a network, a VXLAN tunnel ID, and a VLAN.

Method	URI	Description
GET	/v2.0/networks	Lists networks that are accessible to the tenant who submits the request. Networks with multiple segments include the segments list in the response.
POST	/v2.0/networks	Creates a network with multiple segment mappings.
GET	/v2.0/networks/{network_id}	Shows details for a network with multiple segments.

15.8.1. List networks

Method	URI	Description
GET	/v2.0/networks	Lists networks that are accessible to the tenant who submits the request. Networks with multiple segments include the segments list in the response.

Normal response codes: 200

Error response codes: unauthorized (401)

15.8.1.1. Request

This operation does not accept a request body.

15.8.1.2. Response

Example 15.118. List networks: JSON response

```
{
  "networks": [
    {
      "status": "ACTIVE",
      "subnets": [],
      "name": "net1",
      "admin_state_up": true,
      "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
      "segments": [
        {
          "provider:segmentation_id": 2,
          "provider:physical_network": "8bab8453-1bc9-45af-8c70-f83aa9b50453",
          "provider:network_type": "vlan"
        },
        {
          "provider:segmentation_id": 0,
          "provider:physical_network": "8bab8453-1bc9-45af-8c70-f83aa9b50453",
          "provider:network_type": "stt"
        }
      ],
      "router:external": false,
      "shared": false,
      "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c"
    },
    {
      "status": "ACTIVE",
      "subnets": [
        "08eae331-0402-425a-923c-34f7cfe39c1b"
      ],
      "name": "private",
      "provider:physical_network": null,
      "router:external": true,
      "admin_state_up": true,
      "tenant_id": "26a7980765d0414dbc1fc1f88cdb7e6e",
      "provider:network_type": "local",
    }
  ]
}
```

```
        "shared": true,  
        "id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",  
        "provider:segmentation_id": null  
      }  
    ]  
  }
```

15.8.2. Create network with multiple segment mappings

Method	URI	Description
POST	/v2.0/networks	Creates a network with multiple segment mappings.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

15.8.2.1. Request

Example 15.119. Create network with multiple segment mappings: JSON request

```
{
  "network": {
    "segments": [
      {
        "provider:segmentation_id": "2",
        "provider:physical_network": "8bab8453-1bc9-45af-8c70-f83aa9b50453",
        "provider:network_type": "vlan"
      },
      {
        "provider:physical_network": "8bab8453-1bc9-45af-8c70-f83aa9b50453",
        "provider:network_type": "stt"
      }
    ],
    "name": "net1",
    "admin_state_up": true
  }
}
```

15.8.2.2. Response

Example 15.120. Create network with multiple segment mappings: JSON response

```
{
  "network": {
    "status": "ACTIVE",
    "subnets": [],
    "name": "net1",
    "admin_state_up": true,
    "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
    "segments": [
      {
        "provider:segmentation_id": 2,
        "provider:physical_network": "8bab8453-1bc9-45af-8c70-f83aa9b50453",
        "provider:network_type": "vlan"
      },
      {
        "provider:segmentation_id": null,
```



```
        "provider:physical_network": "8bab8453-1bc9-45af-8c70-  
f83aa9b50453",  
        "provider:network_type": "stt"  
    },  
    ],  
    "shared": false,  
    "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c"  
}
```

15.8.3. Show details for a network with multiple segments

Method	URI	Description
GET	/v2.0/networks/{network_id}	Shows details for a network with multiple segments.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

15.8.3.1. Request

This table shows the URI parameters for the show details for a network with multiple segments request:

Name	Type	Description
{network_id}	UUID	The UUID of the network.

This operation does not accept a request body.

15.8.3.2. Response

Example 15.121. Show details for a network with multiple segments: JSON response

```
{
  "network": {
    "status": "ACTIVE",
    "subnets": [],
    "name": "net1",
    "admin_state_up": true,
    "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
    "segments": [
      {
        "provider:segmentation_id": 2,
        "provider:physical_network": "8bab8453-1bc9-45af-8c70-f83aa9b50453",
        "provider:network_type": "vlan"
      },
      {
        "provider:segmentation_id": 0,
        "provider:physical_network": "8bab8453-1bc9-45af-8c70-f83aa9b50453",
        "provider:network_type": "stt"
      }
    ],
    "router:external": false,
    "shared": false,
    "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c"
  }
}
```

15.9. VLAN transparency extension (networks)

Enables plug-ins that support VLAN transparency to deliver VLAN-transparent trunk networks. If the service does not support VLAN transparency and a user requests a VLAN-trans-

parent network, the plug-in refuses to create one and returns an appropriate error to the user.

You cannot update the `vlan-transparent` attribute. If you try to do so, an error occurs.

To delete a VLAN-transparent network, issue a normal delete network request.

The ML2 plug-in currently supports this extension. With the ML2 plug-in, you can set the `vlan-transparent` attribute to either `true` or `false`.

Method	URI	Description
GET	<code>/v2.0/networks</code>	Lists networks. The response shows the VLAN transparency attribute.
POST	<code>/v2.0/networks</code>	Creates a VLAN-transparent network.
GET	<code>/v2.0/networks/{network_id}</code>	Shows details for a VLAN-transparent network.

15.9.1. List networks with VLAN transparency attribute

Method	URI	Description
GET	/v2.0/networks	Lists networks. The response shows the VLAN transparency attribute.

Normal response codes: 200

Error response codes: unauthorized (401)

15.9.1.1. Request

This operation does not accept a request body.

15.9.1.2. Response

Example 15.122. List networks with VLAN transparency attribute: JSON response

```
{
  "networks": [
    {
      "status": "ACTIVE",
      "subnets": [],
      "name": "net1",
      "admin_state_up": true,
      "tenant_id": "e252a863-92ee-480f-8bd8-71be77089499",
      "shared": false,
      "router:external": false,
      "vlan_transparent": true,
      "id": "f5e6d63c-04a4-4b2c-8b27-a9854412d5a7"
    },
    {
      "status": "ACTIVE",
      "subnets": [
        "3daba37a-bced-4153-a4bb-d83dcc0552d9"
      ],
      "name": "private",
      "admin_state_up": true,
      "tenant_id": "109e5fae-d976-4791-84c7-6ae0bb3896c3",
      "shared": true,
      "router:external": false,
      "vlan_transparent": false,
      "id": "37e11503-3244-49f1-b92a-9f21bab017d9"
    }
  ]
}
```

15.9.2. Create VLAN-transparent network

Method	URI	Description
POST	/v2.0/networks	Creates a VLAN-transparent network.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

15.9.2.1. Request

Example 15.123. Create VLAN-transparent network: JSON request

```
{
  "network": {
    "name": "net1",
    "admin_state_up": true,
    "vlan_transparent": true
  }
}
```

15.9.2.2. Response

Example 15.124. Create VLAN-transparent network: JSON response

```
{
  "network": {
    "status": "ACTIVE",
    "subnets": [],
    "name": "net1",
    "admin_state_up": true,
    "vlan_transparent": true,
    "tenant_id": "5831268f-1f52-49a7-88d5-bc0d7a74d523",
    "router:external": false,
    "shared": false,
    "id": "3114f6e9-f9bc-4570-a941-7329b3b9759f"
  }
}
```

15.9.3. Show VLAN-transparent network details

Method	URI	Description
GET	/v2.0/networks/{network_id}	Shows details for a VLAN-transparent network.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

15.9.3.1. Request

This table shows the URI parameters for the show vlan-transparent network details request:

Name	Type	Description
{network_id}	UUID	The UUID of the network.

This operation does not accept a request body.

15.9.3.2. Response

Example 15.125. Show VLAN-transparent network details: JSON response

```
{
  "network": {
    "status": "ACTIVE",
    "subnets": [],
    "name": "net1",
    "admin_state_up": true,
    "tenant_id": "e926fd5a-e9f6-4dc8-8043-a352d974ceaf",
    "router:external": false,
    "vlan_transparent": true,
    "shared": false,
    "id": "20403fe9-6c9c-48e5-9edb-c3426a955068"
  }
}
```

15.10. Ports binding extended attributes (ports)

Lists, creates, shows information for, updates, and deletes ports.

Method	URI	Description
GET	/v2.0/ports{?status,display_name,admin_state,network_id,tenant_id,device_owner,mac_address,port_id,security_groups,device_id}	Lists ports to which the tenant has access.
POST	/v2.0/ports	Creates a port on a network.
GET	/v2.0/ports/{port_id}	Shows details for a port.
PUT	/v2.0/ports/{port_id}	Updates a port.
DELETE	/v2.0/ports/{port_id}	Deletes a port.

15.10.1. List ports

Method	URI	Description
GET	/v2.0/ports{?status,display_name,admin_state,network_id,tenant_id,device_owner,mac_address,port_id,security_groups,device_id}	Lists ports to which the tenant has access.

Normal response codes: 200

Error response codes: unauthorized (401)

15.10.1.1. Request

This table shows the query parameters for the list ports request:

Name	Type	Description
status	String (Optional)	The port status. Value is ACTIVE or DOWN.
display_name	String (Optional)	The port name.
admin_state	Boolean (Optional)	The administrative state of the router, which is up (true) or down (false).
network_id	UUID (Optional)	The UUID of the attached network.
tenant_id	UUID (Optional)	The UUID of the tenant who owns the network. Only administrative users can specify a tenant UUID other than their own. You cannot change this value through authorization policies.
device_owner	String (Optional)	The UUID of the entity that uses this port. For example, a DHCP agent.
mac_address	String (Optional)	The MAC address of the port.
port_id	UUID (Optional)	The UUID of the port.
security_groups	List (Optional)	The UUIDs of any attached security groups.
device_id	UUID (Optional)	The UUID of the device that uses this port. For example, a virtual server.

15.10.1.2. Response

Example 15.126. List ports: JSON response

```
{
  "ports": [
    {
      "status": "ACTIVE",
```

```

        "name": "",
        "allowed_address_pairs": [],
        "admin_state_up": true,
        "network_id": "70c1db1f-b701-45bd-96e0-a313ee3430b3",
        "tenant_id": "",
        "extra_dhcp_opts": [],
        "device_owner": "network:router_gateway",
        "mac_address": "fa:16:3e:58:42:ed",
        "fixed_ips": [
            {
                "subnet_id": "008ba151-0b8c-4a67-98b5-0d2b87666062",
                "ip_address": "172.24.4.2"
            }
        ],
        "id": "d80b1a3b-4fc1-49f3-952e-1e2ab7081d8b",
        "security_groups": [],
        "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824"
    },
    {
        "status": "ACTIVE",
        "name": "",
        "allowed_address_pairs": [],
        "admin_state_up": true,
        "network_id": "f27aa545-cbdd-4907-b0c6-c9e8b039dcc2",
        "tenant_id": "d397de8a63f341818f198abb0966f6f3",
        "extra_dhcp_opts": [],
        "device_owner": "network:router_interface",
        "mac_address": "fa:16:3e:bb:3c:e4",
        "fixed_ips": [
            {
                "subnet_id": "288bf4a1-51ba-43b6-9d0a-520e9005db17",
                "ip_address": "10.0.0.1"
            }
        ],
        "id": "f71a6703-d6de-4be1-a91a-a570ede1d159",
        "security_groups": [],
        "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824"
    }
]
}

```


15.10.2. Create port

Method	URI	Description
POST	/v2.0/ports	Creates a port on a network.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), forbidden (403)

15.10.2.1. Request

Example 15.127. Create port: JSON request

```
{
  "port": {
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "name": "private-port",
    "admin_state_up": true
  }
}
```

15.10.2.2. Response

Example 15.128. Create port: JSON response

```
{
  "port": {
    "status": "DOWN",
    "binding:host_id": "",
    "name": "private-port",
    "allowed_address_pairs": [],
    "admin_state_up": true,
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "tenant_id": "d6700c0c9ffa4f1cb322cd4a1f3906fa",
    "binding:vif_details": {},
    "binding:vnic_type": "normal",
    "binding:vif_type": "unbound",
    "device_owner": "",
    "mac_address": "fa:16:3e:c9:cb:f0",
    "binding:profile": {},
    "fixed_ips": [
      {
        "subnet_id": "a0304c3a-4f08-4c43-88af-d796509c97d2",
        "ip_address": "10.0.0.2"
      }
    ],
    "id": "65c0ee9f-d634-4522-8954-51021b570b0d",
    "security_groups": [
      "f0ac4394-7e4a-4409-9701-ba8be283dbc3"
    ],
    "device_id": ""
  }
}
```

15.10.3. Show port details

Method	URI	Description
GET	/v2.0/ports/{port_id}	Shows details for a port.

Normal response codes: 200

15.10.3.1. Request

This table shows the URI parameters for the show port details request:

Name	Type	Description
{port_id}	UUID	The UUID of the port.

This operation does not accept a request body.

15.10.3.2. Response

Example 15.129. Show port details: JSON response

```
{
  "port": {
    "status": "ACTIVE",
    "binding:host_id": "devstack",
    "name": "",
    "allowed_address_pairs": [],
    "admin_state_up": true,
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "tenant_id": "7e02058126cc4950b75f9970368ba177",
    "extra_dhcp_opts": [],
    "binding:vif_details": {
      "port_filter": true,
      "ovs_hybrid_plug": true
    },
    "binding:vif_type": "ovs",
    "device_owner": "network:router_interface",
    "port_security_enabled": false,
    "mac_address": "fa:16:3e:23:fd:d7",
    "binding:profile": {},
    "binding:vnic_type": "normal",
    "fixed_ips": [
      {
        "subnet_id": "a0304c3a-4f08-4c43-88af-d796509c97d2",
        "ip_address": "10.0.0.1"
      }
    ],
    "id": "46d4bfb9-b26e-41f3-bd2e-e6dcc1ccedb2",
    "security_groups": [],
    "device_id": "5e3898d7-11be-483e-9732-b2f5eccd2b2e"
  }
}
```

15.10.4. Update port

Method	URI	Description
PUT	/v2.0/ports/{port_id}	Updates a port.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.10.4.1. Request

This table shows the URI parameters for the update port request:

Name	Type	Description
{port_id}	UUID	The UUID of the port.

Example 15.130. Update port: JSON request

```
{
  "port": {
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "name": "private-port",
    "admin_state_up": true
  }
}
```

15.10.4.2. Response

Example 15.131. Update port: JSON response

```
{
  "port": {
    "status": "DOWN",
    "binding:host_id": "",
    "name": "private-port",
    "allowed_address_pairs": [],
    "admin_state_up": true,
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "tenant_id": "d6700c0c9ffa4f1cb322cd4a1f3906fa",
    "binding:vif_details": {},
    "binding:vnic_type": "normal",
    "binding:vif_type": "unbound",
    "device_owner": "",
    "mac_address": "fa:16:3e:c9:cb:f0",
    "binding:profile": {},
    "fixed_ips": [
      {
        "subnet_id": "a0304c3a-4f08-4c43-88af-d796509c97d2",
        "ip_address": "10.0.0.2"
      }
    ],
    "id": "65c0ee9f-d634-4522-8954-51021b570b0d",
    "security_groups": [
      "f0ac4394-7e4a-4409-9701-ba8be283dbc3"
    ]
  }
}
```

```
    "device_id": ""  
  }  
}
```

15.10.5. Delete port

Method	URI	Description
DELETE	/v2.0/ports/{port_id}	Deletes a port.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.10.5.1. Request

This table shows the URI parameters for the delete port request:

Name	Type	Description
{port_id}	UUID	The UUID of the port.

This operation does not accept a request body.

15.11. QoS policies (qos)

Lists, creates, shows information for, and updates QoS policies.

Method	URI	Description
GET	/v2.0/qos/policies	Lists all QoS policies that are associated with your tenant account.
POST	/v2.0/qos/policies	Creates a QoS policy.
GET	/v2.0/qos/policies/{policy_id}	Shows details for a QoS policy.
PUT	/v2.0/qos/policies/{policy_id}	Updates a QoS policy.
DELETE	/v2.0/qos/policies/{policy_id}	Deletes a QoS policy.
GET	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules	Lists all bandwidth limit rules for a QoS policy.
POST	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules	Creates a bandwidth limit rule for a QoS policy.
GET	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules/{rule_id}	Shows details for a bandwidth limit rule for a QoS policy.
PUT	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules/{rule_id}	Updates a bandwidth limit rule for a QoS policy.
DELETE	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules/{rule_id}	Deletes a bandwidth limit rule for a QoS policy.

15.11.1. List QoS policies

Method	URI	Description
GET	/v2.0/qos/policies	Lists all QoS policies that are associated with your tenant account.

The list might be empty.

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

15.11.1.1. Request

This operation does not accept a request body.

15.11.1.2. Response

Example 15.132. List QoS policies: JSON response

```
{
  "policies": [
    {
      "tenant_id": "8d4c70a21fed4aeba121a1a429ba0d04",
      "id": "46ebaec0-0570-43ac-82f6-60d2b03168c4",
      "name": "10Mbit",
      "description": "This policy limits the ports to 10Mbit max.",
      "shared": false,
      "bandwidth_limit_rules": [
        {
          "id": "5f126d84-551a-4dcf-bb01-0e9c0df0c793",
          "policy_id": "46ebaec0-0570-43ac-82f6-60d2b03168c4",
          "max_kbps": "10000",
          "max_burst_kbps": "0"
        }
      ]
    }
  ]
}
```

15.11.2. Create QoS policy

Method	URI	Description
POST	/v2.0/qos/policies	Creates a QoS policy.

Creates a QoS policy by using the configuration that you define in the request object. A response object is returned. The object contains a unique ID.

If the caller is not an administrative user, this call returns the HTTP Forbidden (403) response code.

Users with an administrative role can create policies on behalf of other tenants by specifying a tenant UUID that is different than their own.

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.11.2.1. Request

Example 15.133. Create QoS policy: JSON request

```
{
  "policy": {
    "name": "10Mbit",
    "description": "This policy limits the ports to 10Mbit max.",
    "shared": false
  }
}
```

15.11.2.2. Response

Example 15.134. Create QoS policy: JSON response

```
{
  "policy": {
    "name": "10Mbit",
    "description": "This policy limits the ports to 10Mbit max.",
    "id": "46ebaec0-0570-43ac-82f6-60d2b03168c4",
    "tenant_id": "8d4c70a21fed4aeba121a1a429ba0d04",
    "shared": false
  }
}
```

15.11.3. Show QoS policy details

Method	URI	Description
GET	/v2.0/qos/policies/{policy_id}	Shows details for a QoS policy.

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

15.11.3.1. Request

This table shows the URI parameters for the show qos policy details request:

Name	Type	Description
{policy_id}	UUID	The UUID of the policy.

This operation does not accept a request body.

15.11.3.2. Response

Example 15.135. Show QoS policy details: JSON response

```
{
  "policy": {
    "tenant_id": "8d4c70a21fed4aeba121a1a429ba0d04",
    "id": "46ebaec0-0570-43ac-82f6-60d2b03168c4",
    "name": "10Mbit",
    "description": "This policy limits the ports to 10Mbit max.",
    "shared": false,
    "bandwidth_limit_rules": [
      {
        "id": "5f126d84-551a-4dcf-bb01-0e9c0df0c793",
        "policy_id": "46ebaec0-0570-43ac-82f6-60d2b03168c4",
        "max_kbps": "10000",
        "max_burst_kbps": "0"
      }
    ]
  }
}
```


15.11.4. Update QoS policy

Method	URI	Description
PUT	/v2.0/qos/policies/{policy_id}	Updates a QoS policy.

If the request is valid, the service returns the Accepted (202) response code.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.11.4.1. Request

This table shows the URI parameters for the update qos policy request:

Name	Type	Description
{policy_id}	UUID	The UUID of the policy.

Example 15.136. Update QoS policy: JSON request

```
{
  "policy": {
    "name": "10Mbit",
    "description": "This policy limits the ports to 10Mbit max.",
    "shared": false
  }
}
```

15.11.4.2. Response

Example 15.137. Update QoS policy: JSON response

```
{
  "policy": {
    "name": "10Mbit",
    "description": "This policy limits the ports to 10Mbit max.",
    "id": "46ebaec0-0570-43ac-82f6-60d2b03168c4",
    "tenant_id": "8d4c70a21fed4aeba121a1a429ba0d04",
    "shared": false
  }
}
```

15.11.5. Delete QoS policy

Method	URI	Description
DELETE	/v2.0/qos/policies/{policy_id}	Deletes a QoS policy.

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.11.5.1. Request

This table shows the URI parameters for the delete qos policy request:

Name	Type	Description
{policy_id}	UUID	The UUID of the policy.

This operation does not accept a request body.

15.11.6. List bandwidth limit rules for QoS policy

Method	URI	Description
GET	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules	Lists all bandwidth limit rules for a QoS policy.

The list might be empty.

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

15.11.6.1. Request

This table shows the URI parameters for the list bandwidth limit rules for qos policy request:

Name	Type	Description
{policy_id}	UUID	The UUID of the policy.

This operation does not accept a request body.

15.11.6.2. Response

Example 15.138. List bandwidth limit rules for QoS policy: JSON response

```
{
  "bandwidth_limit_rules": [
    {
      "id": "5f126d84-551a-4dcf-bb01-0e9c0df0c793",
      "policy_id": "46ebaec0-0570-43ac-82f6-60d2b03168c4",
      "max_kbps": "10000",
      "max_burst_kbps": "0"
    }
  ]
}
```

15.11.7. Create bandwidth limit rule

Method	URI	Description
POST	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules	Creates a bandwidth limit rule for a QoS policy.

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.11.7.1. Request

This table shows the URI parameters for the create bandwidth limit rule request:

Name	Type	Description
{policy_id}	UUID	The UUID of the policy.

Example 15.139. Create bandwidth limit rule: JSON request

```
{
  "bandwidth_limit_rule": {
    "max_kbps": "10000"
  }
}
```

15.11.7.2. Response

Example 15.140. Create bandwidth limit rule: JSON response

```
{
  "bandwidth_limit_rule": {
    "id": "5f126d84-551a-4dcf-bb01-0e9c0df0c793",
    "policy_id": "46ebaec0-0570-43ac-82f6-60d2b03168c4",
    "max_kbps": "10000",
    "max_burst_kbps": "0"
  }
}
```

15.11.8. Show bandwidth limit rule details

Method	URI	Description
GET	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules/{rule_id}	Shows details for a bandwidth limit rule for a QoS policy.

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

15.11.8.1. Request

This table shows the URI parameters for the show bandwidth limit rule details request:

Name	Type	Description
{policy_id}	UUID	The UUID of the policy.
{rule_id}	UUID	The UUID of the rule.

This operation does not accept a request body.

15.11.8.2. Response

Example 15.141. Show bandwidth limit rule details: JSON response

```
{
  "bandwidth_limit_rule": {
    "id": "5f126d84-551a-4dcf-bb01-0e9c0df0c793",
    "policy_id": "46ebaec0-0570-43ac-82f6-60d2b03168c4",
    "max_kbps": "10000",
    "max_burst_kbps": "0"
  }
}
```

15.11.9. Update bandwidth limit rule

Method	URI	Description
PUT	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules/{rule_id}	Updates a bandwidth limit rule for a QoS policy.

If the request is valid, the service returns the Accepted (202) response code.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.11.9.1. Request

This table shows the URI parameters for the update bandwidth limit rule request:

Name	Type	Description
{policy_id}	UUID	The UUID of the policy.
{rule_id}	UUID	The UUID of the rule.

Example 15.142. Update bandwidth limit rule: JSON request

```
{
  "bandwidth_limit_rule": {
    "max_kbps": "10000"
  }
}
```

15.11.9.2. Response

Example 15.143. Update bandwidth limit rule: JSON response

```
{
  "bandwidth_limit_rule": {
    "id": "5f126d84-551a-4dcf-bb01-0e9c0df0c793",
    "policy_id": "46ebaec0-0570-43ac-82f6-60d2b03168c4",
    "max_kbps": "10000",
    "max_burst_kbps": "0"
  }
}
```

15.11.10. Delete bandwidth limit rule

Method	URI	Description
DELETE	/v2.0/qos/policies/{policy_id}/bandwidth_limit_rules/{rule_id}	Deletes a bandwidth limit rule for a QoS policy.

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

15.11.10.1. Request

This table shows the URI parameters for the delete bandwidth limit rule request:

Name	Type	Description
{policy_id}	UUID	The UUID of the policy.
{rule_id}	UUID	The UUID of the rule.

This operation does not accept a request body.

15.12. Quotas extension (quotas)

Lists quotas for tenants who have non-default quota values, and lists, updates, and resets quotas for a tenant.

A quota value of -1 means that quota has no limit.

Method	URI	Description
GET	/v2.0/quotas	Lists quotas for tenants who have non-default quota values.
GET	/v2.0/quotas/{tenant_id}	Lists quotas for a tenant.
PUT	/v2.0/quotas/{tenant_id}	Updates quotas for a tenant. Use when non-default quotas are desired.
DELETE	/v2.0/quotas/{tenant_id}	Resets quotas to default values for a tenant.

15.12.1. List quotas for tenants with non-default quota values

Method	URI	Description
GET	/v2.0/quotas	Lists quotas for tenants who have non-default quota values.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

15.12.1.1. Request

This operation does not accept a request body.

15.12.1.2. Response

Example 15.144. List quotas for tenants with non-default quota values: JSON response

```
{
  "quotas": [
    {
      "subnet": 10,
      "network": 15,
      "floatingip": 50,
      "tenant_id": "bab7d5c60cd041a0a36f7c4b6e1dd978",
      "subnetpool": -1,
      "security_group_rule": 100,
      "security_group": 10,
      "router": 10,
      "rbac_policy": -1,
      "port": 50
    }
  ]
}
```


15.12.2. List quotas for a tenant

Method	URI	Description
GET	/v2.0/quotas/{tenant_id}	Lists quotas for a tenant.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.12.2.1. Request

This table shows the URI parameters for the list quotas for a tenant request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant.

This operation does not accept a request body.

15.12.2.2. Response

Example 15.145. List quotas for a tenant: JSON response

```
{
  "quota": {
    "subnet": 10,
    "network": 10,
    "floatingip": 50,
    "subnetpool": -1,
    "security_group_rule": 100,
    "security_group": 10,
    "router": 10,
    "rbac_policy": -1,
    "port": 50
  }
}
```

15.12.3. Update quota for a tenant

Method	URI	Description
PUT	/v2.0/quotas/{tenant_id}	Updates quotas for a tenant. Use when non-default quotas are desired.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.12.3.1. Request

This table shows the URI parameters for the update quota for a tenant request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant.

Example 15.146. Update quota for a tenant: JSON request

```
{
  "quota": {
    "subnet": 10,
    "network": 10,
    "floatingip": 50,
    "subnetpool": -1,
    "security_group_rule": 100,
    "security_group": 10,
    "router": 10,
    "rbac_policy": -1,
    "port": 50
  }
}
```

15.12.3.2. Response

Example 15.147. Update quota for a tenant: JSON response

```
{
  "quota": {
    "subnet": 10,
    "network": 15,
    "floatingip": 50,
    "subnetpool": -1,
    "security_group_rule": 100,
    "security_group": 10,
    "router": 10,
    "rbac_policy": -1,
    "port": 50
  }
}
```

15.12.4. Reset quota for a tenant

Method	URI	Description
DELETE	/v2.0/quotas/{tenant_id}	Resets quotas to default values for a tenant.

Normal response codes: 204

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.12.4.1. Request

This table shows the URI parameters for the reset quota for a tenant request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant.

This operation does not accept a request body.

15.13. Extra routes (routers)

Adds extra routes to the router resource.

You can update a router to add a set of next hop IPs and destination CIDRs.



Note

The next hop IP must be part of a subnet to which the router interfaces are connected. You can configure the routes attribute on only update operations.

Method	URI	Description
PUT	/v2.0/routers/{router_id}	Updates extra routes on a router.

15.13.1. Update extra routes

Method	URI	Description
PUT	/v2.0/routers/{router_id}	Updates extra routes on a router.

The next hop IP address must be a part of one of the subnets to which the router interfaces are connected. Otherwise, the server responds with the **Bad Request (400)** error code.

When a validation error is detected, such as a format error of IP address or CIDR, the server responds with the **Bad Request (400)** response code.

When Networking receives a request to delete the router interface for subnets that are used by one or more routes, it responds with a **Conflict (409)** response code.

Normal response codes: 200

Error response codes: **badRequest (400)**, **unauthorized (401)**, **itemNotFound (404)**, **conflict (409)**

15.13.1.1. Request

This table shows the URI parameters for the update extra routes request:

Name	Type	Description
{router_id}	UUID	The UUID of the router.

Example 15.148. Update extra routes: JSON request

```
{
  "router": {
    "external_gateway_info": {
      "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b",
      "enable_snat": true,
      "external_fixed_ips": [
        {
          "subnet_id": "255.255.255.0",
          "ip": "192.168.10.1"
        }
      ]
    }
  }
}
```

15.13.1.2. Response

Example 15.149. Update extra routes: JSON response

```
{
  "router": {
    "status": "ACTIVE",
    "external_gateway_info": {
      "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b",
      "enable_snat": true,
      "external_fixed_ips": [
```

```

        {
            "subnet_id": "255.255.255.0",
            "ip": "192.168.10.2"
        }
    ],
    "name": "another_router",
    "admin_state_up": true,
    "tenant_id": "6b96ff0cb17a4b859e1e575d221683d3",
    "distributed": false,
    "routes": [],
    "ha": false,
    "id": "8604a0de-7f6b-409a-a47c-a1cc7bc77b2e"
}

```

15.14. Layer-3 networking (routers, floatingips)

Routes packets between subnets, forwards packets from internal networks to external ones, and accesses instances from external networks through floating IPs.

This extension introduces these resources:

- **router**. A logical entity for forwarding packets across internal subnets and NATting them on external networks through an appropriate external gateway.
- **floatingip**. An external IP address that you map to a port in an internal network.

Method	URI	Description
GET	/v2.0/routers	Lists logical routers that the tenant who submits the request can access.
POST	/v2.0/routers	Creates a logical router.
GET	/v2.0/routers/{router_id}	Shows details for a router.
PUT	/v2.0/routers/{router_id}	Updates a logical router.
DELETE	/v2.0/routers/{router_id}	Deletes a logical router and, if present, its external gateway interface.
PUT	/v2.0/routers/{router_id}/add_router_interface	Adds an internal interface to a logical router.
PUT	/v2.0/routers/{router_id}/remove_router_interface	Deletes an internal interface from a logical router.
GET	/v2.0/floatingips	Lists floating IPs that are accessible to the tenant who submits the request.
POST	/v2.0/floatingips	Creates a floating IP, and, if you specify port information, associates the floating IP with an internal port.
GET	/v2.0/floatingips/{floatingip_id}	Shows details for a floating IP.
PUT	/v2.0/floatingips/{floatingip_id}	Updates a floating IP and its association with an internal port.
DELETE	/v2.0/floatingips/{floatingip_id}	Deletes a floating IP and, if present, its associated port.

15.14.1. List routers

Method	URI	Description
GET	/v2.0/routers	Lists logical routers that the tenant who submits the request can access.

Default policy settings return only those routers that the tenant who submits the request owns, unless an administrative user submits the request.

This example request lists routers in JSON format:

```
GET /v2.0/routers Accept: application/json
```

Use the `fields` query parameter to control which fields are returned in the response body. Additionally, you can filter results by using query string parameters. For information, see [Filtering and Column Selection](#).

Normal response codes: 200

Error response codes: unauthorized (401)

15.14.1.1. Request

This operation does not accept a request body.

15.14.1.2. Response

Example 15.150. List routers: JSON response

```
{
  "routers": [
    {
      "status": "ACTIVE",
      "external_gateway_info": null,
      "name": "second_routers",
      "admin_state_up": true,
      "tenant_id": "6b96ff0cb17a4b859e1e575d221683d3",
      "routes": [],
      "id": "7177abc4-5ae9-4bb7-b0d4-89e94a4abf3b"
    },
    {
      "status": "ACTIVE",
      "external_gateway_info": {
        "network_id": "3c5bcddd-6af9-4e6b-9c3e-c153e521cab8",
        "enable_snat": true,
        "external_fixed_ips": [
          {
            "subnet_id": "255.255.255.0",
            "ip": "192.168.10.2"
          }
        ]
      },
      "name": "router1",
      "admin_state_up": true,
      "tenant_id": "33a40233088643acb66ff6eb0ebea679",
    }
  ]
}
```

```
        "routes": [],  
        "id": "a9254bdb-2613-4a13-ac4c-adc581fba50d"  
    }  
]  
}
```

15.14.2. Create router

Method	URI	Description
POST	/v2.0/routers	Creates a logical router.

This operation creates a logical router. The logical router does not have any internal interface and it is not associated with any subnet. You can optionally specify an external gateway for a router at create time. The external gateway for the router must be plugged into an external network. An external network has its `router : external` extended field set to `true`. To specify an external gateway, the UUID of the external network must be passed in the `external_gateway_info` attribute in the request body, as follows:

```
{
  "router": {
    "external_gateway_info": {
      "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b"
    }
  }
}
```

Normal response codes: 201

Error response codes: `badRequest (400)`, `unauthorized (401)`

15.14.2.1. Request

Example 15.151. Create router: JSON request

```
{
  "router": {
    "name": "another_router",
    "external_gateway_info": {
      "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b",
      "enable_snat": true,
      "external_fixed_ips": [
        {
          "subnet_id": "255.255.255.0",
          "ip": "192.168.10.1"
        }
      ]
    },
    "admin_state_up": true
  }
}
```

15.14.2.2. Response

Example 15.152. Create router: JSON response

```
{
  "router": {
    "status": "ACTIVE",
    "external_gateway_info": {
      "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b",
      "enable_snat": true,

```



```
        "external_fixed_ips": [  
            {  
                "subnet_id": "255.255.255.0",  
                "ip": "192.168.10.2"  
            }  
        ],  
        "name": "another_router",  
        "admin_state_up": true,  
        "tenant_id": "6b96ff0cb17a4b859e1e575d221683d3",  
        "distributed": false,  
        "routes": [],  
        "ha": false,  
        "id": "8604a0de-7f6b-409a-a47c-a1cc7bc77b2e"  
    }  
}
```

15.14.3. Show router details

Method	URI	Description
GET	/v2.0/routers/{router_id}	Shows details for a router.

This example request shows details for a router in JSON format:

```
GET /v2.0/routers/{router_id} Accept: application/json
```

Use the `fields` query parameter to control which fields are returned in the response body. For information, see [Filtering and Column Selection](#).

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.14.3.1. Request

This table shows the URI parameters for the show router details request:

Name	Type	Description
{router_id}	UUID	The UUID of the router.

This operation does not accept a request body.

15.14.3.2. Response

Example 15.153. Show router details: JSON response

```
{
  "router": {
    "status": "ACTIVE",
    "external_gateway_info": {
      "network_id": "85d76829-6415-48ff-9c63-5c5ca8c61ac6",
      "enable_snat": true,
      "external_fixed_ips": [
        {
          "subnet_id": "255.255.255.0",
          "ip": "192.168.10.2"
        }
      ]
    },
    "name": "router1",
    "admin_state_up": true,
    "tenant_id": "d6554fe62e2f41efbb6e026fad5c1542",
    "routes": [],
    "id": "a07eea83-7710-4860-931b-5fe220fae533"
  }
}
```

15.14.4. Update router

Method	URI	Description
PUT	/v2.0/routers/{router_id}	Updates a logical router.

You can update the name, administrative state, and the external gateway. For more information about how to set the external gateway for a router, see the create router operation. This operation does not enable the update of router interfaces. To update a router, use the add router interface and remove router interface operations.

This example updates the external gateway information for a router:

```
PUT /v2.0/routers/{router_id} Accept: application/json
```

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.14.4.1. Request

This table shows the URI parameters for the update router request:

Name	Type	Description
{router_id}	UUID	The UUID of the router.

Example 15.154. Update router: JSON request

```
{
  "router": {
    "external_gateway_info": {
      "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b",
      "enable_snat": true,
      "external_fixed_ips": [
        {
          "subnet_id": "255.255.255.0",
          "ip": "192.168.10.1"
        }
      ]
    }
  }
}
```

15.14.4.2. Response

Example 15.155. Update router: JSON response

```
{
  "router": {
    "status": "ACTIVE",
    "external_gateway_info": {
      "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b",
      "enable_snat": true,
      "external_fixed_ips": [
        {

```

```
        "subnet_id": "255.255.255.0",
        "ip": "192.168.10.2"
      }
    ]
  },
  "name": "another_router",
  "admin_state_up": true,
  "tenant_id": "6b96ff0cb17a4b859e1e575d221683d3",
  "distributed": false,
  "routes": [],
  "ha": false,
  "id": "8604a0de-7f6b-409a-a47c-a1cc7bc77b2e"
}
```

15.14.5. Delete router

Method	URI	Description
DELETE	/v2.0/routers/{router_id}	Deletes a logical router and, if present, its external gateway interface.

This operation fails if the router has attached interfaces.

Use the remove router interface operation to remove all router interfaces before you delete the router.

This example deletes a router:

```
DELETE /v2.0/routers/{router_id} Accept: application/json
```

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.14.5.1. Request

This table shows the URI parameters for the delete router request:

Name	Type	Description
{router_id}	UUID	The UUID of the router.

This operation does not accept a request body.

15.14.6. Add interface to router

Method	URI	Description
PUT	/v2.0/routers/{router_id}/add_router_interface	Adds an internal interface to a logical router.

Attaches a subnet to an internal router interface.

Specify the UUID of a subnet or port in the request body:

- Subnet UUID. The gateway IP address for the subnet is used to create the router interface.
- Port UUID. The IP address associated with the port is used to create the router interface.

When you specify an IPv6 subnet, this operation adds the subnet to an existing internal port with same network UUID, on the router. If a port with the same network UUID does not exist, this operation creates a port on the router for that subnet.

The limitation of one IPv4 subnet per router port remains, though a port can contain any number of IPv6 subnets that belong to the same network UUID.

When you use the `port-create` command to add a port and then call `router-interface-add` with this port UUID, this operation adds the port to the router if the following conditions are met:

- The port has no more than one IPv4 subnet.

The IPv6 subnets, if any, on the port do not have same network UUID as the network UUID of IPv6 subnets on any other ports.

If you specify both UUIDs, this operation returns the `BadRequest (400)` response code.

If the port is already in use, this operation returns the `Conflict (409)` response code.

This operation returns a port UUID that is either:

- The same UUID that is passed in the request body.
- The UUID of a port that this operation creates to attach the subnet to the router.

After you run this operation, the operation sets:

- The device UUID of this port to the router UUID.
- The `device_owner` attribute to `network:router_interface`.

Normal response codes: 200

Error response codes: `badRequest (400)`, `unauthorized (401)`, `itemNotFound (404)`, `conflict (409)`

15.14.6.1. Request

This table shows the URI parameters for the add interface to router request:

Name	Type	Description
{router_id}	UUID	The UUID of the router.

Example 15.156. Add interface to router: JSON request

```
{
  "subnet_id": "a2f1f29d-571b-4533-907f-5803ab96ead1"
}
```

15.14.6.2. Response

Example 15.157. Add interface to router: JSON response

```
{
  "subnet_id": "a2f1f29d-571b-4533-907f-5803ab96ead1",
  "tenant_id": "6ba032e4730d42e2ad928f430f5da33e",
  "subnet_ids": [
    "a2f1f29d-571b-4533-907f-5803ab96ead1"
  ],
  "port_id": "3a44f4e5-1694-493a-a1fb-393881c673a4",
  "id": "b0294d7e-7da4-4202-9882-2ab1de9dabc0"
}
```

15.14.7. Delete interface from router

Method	URI	Description
PUT	/v2.0/routers/{router_id}/remove_router_interface	Deletes an internal interface from a logical router.

This operation deletes an internal router interface, which detaches a subnet from the router. If this subnet UUID is the last subnet on the port, this operation deletes the port itself. You must specify either a subnet UUID or port UUID in the request body; the operation uses this value to identify which router interface to deletes.

You can also specify both a subnet UUID and port UUID. If you specify both UUIDs, the subnet UUID must correspond to the subnet UUID of the first IP address on the port. Otherwise, this operation returns the **Conflict** (409) response code with information about the affected router and interface.

If the router or the subnet and port do not exist or are not visible to you, this operation returns the **Not Found** (404) response code. As a consequence of this operation, the operation removes the port connecting the router with the subnet from the subnet for the network.

This example deletes an interface from a router:

```
PUT /v2.0/routers/{router_id}/remove_router_interface Accept: application/json
```

Normal response codes: 200

Error response codes: **badRequest** (400), **unauthorized** (401), **itemNotFound** (404), **conflict** (409)

15.14.7.1. Request

This table shows the URI parameters for the delete interface from router request:

Name	Type	Description
{router_id}	UUID	The UUID of the router.

Example 15.158. Delete interface from router: JSON request

```
{
  "subnet_id": "a2f1f29d-571b-4533-907f-5803ab96ead1"
}
```

15.14.7.2. Response

Example 15.159. Delete interface from router: JSON response

```
{
  "id": "8604a0de-7f6b-409a-a47c-a1cc7bc77b2e",
  "tenant_id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7",
  "port_id": "3a44f4e5-1694-493a-a1fb-393881c673a4",
  "subnet_id": "a2f1f29d-571b-4533-907f-5803ab96ead1"
}
```


15.14.8. List floating IPs

Method	URI	Description
GET	/v2.0/floatingips	Lists floating IPs that are accessible to the tenant who submits the request.

Default policy settings return only those floating IPs that are owned by the tenant who submits the request, unless an admin user submits the request.

This example request lists floating IPs in JSON format:

```
GET /v2.0/floatingips
Accept: application/json
```

Use the `fields` query parameter to control which fields are returned in the response body. Additionally, you can filter results by using query string parameters. For information, see [Filtering and Column Selection](#).

Normal response codes: 200

Error response codes: unauthorized (401)

15.14.8.1. Request

This operation does not accept a request body.

15.14.8.2. Response

Example 15.160. List floating IPs: JSON response

```
{
  "floatingips": [
    {
      "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
      "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
      "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
      "fixed_ip_address": "10.0.0.3",
      "floating_ip_address": "172.24.4.228",
      "port_id": "ce705c24-c1ef-408a-bda3-7bbd946164ab",
      "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7",
      "status": "ACTIVE"
    },
    {
      "router_id": null,
      "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
      "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
      "fixed_ip_address": null,
      "floating_ip_address": "172.24.4.227",
      "port_id": null,
      "id": "61cea855-49cb-4846-997d-801b70c71bdd",
      "status": "DOWN"
    }
  ]
}
```

15.14.9. Create floating IP

Method	URI	Description
POST	/v2.0/floatingips	Creates a floating IP, and, if you specify port information, associates the floating IP with an internal port.

To associate the floating IP with an internal port, specify the port UUID attribute in the request body. If you do not specify a port UUID in the request, you can issue a **PUT** request instead of a **POST** request.

Default policy settings enable only administrative users to set floating IP addresses and some non-administrative users might require a floating IP address. If you do not specify a floating IP address in the request, the operation automatically allocates one.

By default, this operation associates the floating IP address with a single fixed IP address that is configured on an OpenStack Networking port. If a port has multiple IP addresses, you must specify the `fixed_ip_address` attribute in the request body to associate a fixed IP address with the floating IP address.

You can create floating IPs on only external networks. When you create a floating IP, you must specify the UUID of the network on which you want to create the floating IP. Alternatively, you can create a floating IP on a subnet in the external network, based on the costs and quality of that subnet.

You must configure an IP address with the internal OpenStack Networking port that is associated with the floating IP address.

Error codes:

- 400 The operation returns this error code for one of these reasons:
 - The network is not external, such as `router:external=False`.
 - The internal OpenStack Networking port is not associated with the floating IP address.
 - The requested floating IP address does not fall in the subnet range for the external network.
 - The fixed IP address is not valid.
- 401 The operation is not authorized.
- 404 The port UUID is not valid.
- 409 The operation returns this error code for one of these reasons:
 - The requested floating IP address is already in use.
 - The internal OpenStack Networking port and fixed IP address are already associated with another floating IP.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), conflict (409)

15.14.9.1. Request

Example 15.161. Create floating IP: JSON request

```
{
  "floatingip": {
    "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
    "port_id": "ce705c24-c1ef-408a-bda3-7bbd946164ab"
  }
}
```

15.14.9.2. Response

Example 15.162. Create floating IP: JSON response

```
{
  "floatingip": {
    "fixed_ip_address": "10.0.0.3",
    "floating_ip_address": "172.24.4.228",
    "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
    "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7",
    "port_id": "ce705c24-c1ef-408a-bda3-7bbd946164ab",
    "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
    "status": "ACTIVE",
    "tenant_id": "4969c491a3c74ee4af974e6d800c62de"
  }
}
```

15.14.10. Show floating IP details

Method	URI	Description
GET	/v2.0/floatingips/{floatingip_id}	Shows details for a floating IP.

Use the `fields` query parameter to control which fields are returned in the response body. For information, see [Filtering and Column Selection](#).

This example request shows details for a floating IP in JSON format. This example also filters the result by the `fixed_ip_address` and `floating_ip_address` fields.

```
GET /v2.0/floatingips/{floatingip_id}?fields=fixed_ip_address&fields=floating_ip_address
Accept: application/json
```

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.14.10.1. Request

This table shows the URI parameters for the show floating ip details request:

Name	Type	Description
{floatingip_id}	UUID	The UUID of the floating IP address.

This operation does not accept a request body.

15.14.10.2. Response

Example 15.163. Show floating IP details: JSON response

```
{
  "floatingip": {
    "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
    "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
    "fixed_ip_address": "10.0.0.3",
    "floating_ip_address": "172.24.4.228",
    "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
    "status": "ACTIVE",
    "port_id": "ce705c24-c1ef-408a-bda3-7bbd946164ab",
    "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7"
  }
}
```

15.14.11. Update floating IP

Method	URI	Description
PUT	/v2.0/floatingips/{floatingip_id}	Updates a floating IP and its association with an internal port.

The association process is the same as the process for the create floating IP operation.

To disassociate a floating IP from a port, set the `port_id` attribute to null or omit it from the request body.

This example updates a floating IP:

```
PUT /v2.0/floatingips/{floatingip_id} Accept: application/json
```

Depending on the request body that you submit, this request associates a port with or disassociates a port from a floating IP.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), conflict (409)

15.14.11.1. Request

This table shows the URI parameters for the update floating ip request:

Name	Type	Description
{floatingip_id}	UUID	The UUID of the floating IP address.

Example 15.164. Update floating IP (associate port): JSON

```
{
  "floatingip": {
    "port_id": "fc861431-0e6c-4842-a0ed-e2363f9bc3a8"
  }
}
```

Example 15.165. Update floating IP: JSON request

```
{
  "floatingip": {
    "port_id": null
  }
}
```

15.14.11.2. Response

Example 15.166. Update floating IP (associate port): JSON

```
{
  "floatingip": {
    "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
    "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
  }
}
```

```
    "fixed_ip_address": "10.0.0.4",
    "floating_ip_address": "172.24.4.228",
    "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
    "status": "ACTIVE",
    "port_id": "fc861431-0e6c-4842-a0ed-e2363f9bc3a8",
    "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7"
  }
}
```

Example 15.167. Update floating IP: JSON response

```
{
  "floatingip": {
    "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
    "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
    "fixed_ip_address": null,
    "floating_ip_address": "172.24.4.228",
    "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
    "status": "ACTIVE",
    "port_id": null,
    "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7"
  }
}
```

15.14.12. Delete floating IP

Method	URI	Description
DELETE	/v2.0/floatingips/{floatingip_id}	Deletes a floating IP and, if present, its associated port.

This example deletes a floating IP:

```
DELETE /v2.0/floatingips/{floatingip_id} Accept: application/json
```

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404)

15.14.12.1. Request

This table shows the URI parameters for the delete floating ip request:

Name	Type	Description
{floatingip_id}	UUID	The UUID of the floating IP address.

This operation does not accept a request body.

15.15. Security groups (security-groups)

Lists, creates, shows information for, updates, and deletes security groups.

Method	URI	Description
GET	/v2.0/security-groups	Lists OpenStack Networking security groups to which the tenant has access.
POST	/v2.0/security-groups	Creates an OpenStack Networking security group.
GET	/v2.0/security-groups/{security_group_id}{?verbose, fields}	Shows details for a security group.
PUT	/v2.0/security-groups/{security_group_id}	Updates a security group.
DELETE	/v2.0/security-groups/{security_group_id}	Deletes an OpenStack Networking security group.

15.15.1. List security groups

Method	URI	Description
GET	/v2.0/security-groups	Lists OpenStack Networking security groups to which the tenant has access.

The list shows the UUID for and the rules that are associated with each security group.

Normal response codes: 200

Error response codes: unauthorized (401)

15.15.1.1. Request

Example 15.168. List security groups: JSON request

```
GET /v2.0/security-groups
Accept: application/json
```

This operation does not accept a request body.

15.15.1.2. Response

Example 15.169. List security groups: JSON response

```
{
  "security_groups": [
    {
      "description": "default",
      "id": "85cc3048-abc3-43cc-89b3-377341426ac5",
      "name": "default",
      "security_group_rules": [
        {
          "direction": "egress",
          "ethertype": "IPv6",
          "id": "3c0e45ff-adaf-4124-b083-bf390e5482ff",
          "port_range_max": null,
          "port_range_min": null,
          "protocol": null,
          "remote_group_id": null,
          "remote_ip_prefix": null,
          "security_group_id": "85cc3048-
abc3-43cc-89b3-377341426ac5",
          "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
        },
        {
          "direction": "egress",
          "ethertype": "IPv4",
          "id": "93aa42e5-80db-4581-9391-3a608bd0e448",
          "port_range_max": null,
          "port_range_min": null,
          "protocol": null,
          "remote_group_id": null,
          "remote_ip_prefix": null,
          "security_group_id": "85cc3048-
abc3-43cc-89b3-377341426ac5",
```

```
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
      },
      {
        "direction": "ingress",
        "ethertype": "IPv6",
        "id": "c0b09f00-1d49-4e64-a0a7-8a186d928138",
        "port_range_max": null,
        "port_range_min": null,
        "protocol": null,
        "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "remote_ip_prefix": null,
        "security_group_id": "85cc3048-
abc3-43cc-89b3-377341426ac5",
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
      },
      {
        "direction": "ingress",
        "ethertype": "IPv4",
        "id": "f7d45c89-008e-4bab-88ad-d6811724c51c",
        "port_range_max": null,
        "port_range_min": null,
        "protocol": null,
        "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "remote_ip_prefix": null,
        "security_group_id": "85cc3048-
abc3-43cc-89b3-377341426ac5",
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
      }
    ],
    "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
  }
]
```

15.15.2. Create security group

Method	URI	Description
POST	/v2.0/security-groups	Creates an OpenStack Networking security group.

This operation creates a security group with default security group rules for the IPv4 and IPv6 ether types.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

15.15.2.1. Request

Example 15.170. Create security group: JSON request

```
{
  "security_group": {
    "name": "new-webservers",
    "description": "security group for webservers"
  }
}
```

15.15.2.2. Response

Example 15.171. Create security group: JSON response

```
{
  "security_group": {
    "description": "security group for webservers",
    "id": "2076db17-a522-4506-91de-c6dd8e837028",
    "name": "new-webservers",
    "security_group_rules": [
      {
        "direction": "egress",
        "ethertype": "IPv4",
        "id": "38ce2d8e-e8f1-48bd-83c2-d33cb9f50c3d",
        "port_range_max": null,
        "port_range_min": null,
        "protocol": null,
        "remote_group_id": null,
        "remote_ip_prefix": null,
        "security_group_id": "2076db17-a522-4506-91de-c6dd8e837028",
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
      },
      {
        "direction": "egress",
        "ethertype": "IPv6",
        "id": "565b9502-12de-4ffd-91e9-68885cff6ae1",
        "port_range_max": null,
        "port_range_min": null,
        "protocol": null,
        "remote_group_id": null,
        "remote_ip_prefix": null,
        "security_group_id": "2076db17-a522-4506-91de-c6dd8e837028",
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
      }
    ]
  }
}
```

```
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
      },
    ],
    "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
  }
}
```

15.15.3. Show security group

Method	URI	Description
GET	/v2.0/security-groups/{security_group_id}{?verbose, fields}	Shows details for a security group.

The response contains the description, name, UUID, and security group rules that are associated with the security group and tenant.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

15.15.3.1. Request

This table shows the URI parameters for the show security group request:

Name	Type	Description
{security_group_id}	UUID	The UUID of the security group.

This table shows the query parameters for the show security group request:

Name	Type	Description
verbose	Boolean (Optional)	Show detailed information.
fields	String (Optional)	The fields that you want the server to return.

Example 15.172. Show security group: JSON request

```
GET /v2.0/security-groups/85cc3048-abc3-43cc-89b3-377341426ac5
Accept: application/json
```

This operation does not accept a request body.

15.15.3.2. Response

Example 15.173. Show security group: JSON response

```
{
  "security_group": {
    "description": "default",
    "id": "85cc3048-abc3-43cc-89b3-377341426ac5",
    "name": "default",
    "security_group_rules": [
      {
        "direction": "egress",
        "ethertype": "IPv6",
        "id": "3c0e45ff-adaf-4124-b083-bf390e5482ff",
        "port_range_max": null,
        "port_range_min": null,
```

```
        "protocol": null,
        "remote_group_id": null,
        "remote_ip_prefix": null,
        "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
    },
    {
        "direction": "egress",
        "ethertype": "IPv4",
        "id": "93aa42e5-80db-4581-9391-3a608bd0e448",
        "port_range_max": null,
        "port_range_min": null,
        "protocol": null,
        "remote_group_id": null,
        "remote_ip_prefix": null,
        "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
    },
    {
        "direction": "ingress",
        "ethertype": "IPv6",
        "id": "c0b09f00-1d49-4e64-a0a7-8a186d928138",
        "port_range_max": null,
        "port_range_min": null,
        "protocol": null,
        "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "remote_ip_prefix": null,
        "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
    },
    {
        "direction": "ingress",
        "ethertype": "IPv4",
        "id": "f7d45c89-008e-4bab-88ad-d6811724c51c",
        "port_range_max": null,
        "port_range_min": null,
        "protocol": null,
        "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "remote_ip_prefix": null,
        "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
    }
],
"tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
}
```

15.15.4. Update security group

Method	URI	Description
PUT	/v2.0/security-groups/{security_group_id}	Updates a security group.

Normal response codes: 200

Error response codes: serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

15.15.4.1. Request

This table shows the URI parameters for the update security group request:

Name	Type	Description
{security_group_id}	UUID	The UUID of the security group.

Example 15.174. Update security group: JSON request

```
{
  "security_group": {
    "name": "mysecgroup",
    "description": "my security group"
  }
}
```

15.15.4.2. Response

Example 15.175. Update security group: JSON response

```
{
  "security_group": {
    "rules": [],
    "tenant_id": "a52cdb9cc7854a39a23d3af73a40899e",
    "id": "01fbade5-b664-42f6-83ae-4e214f4263fa",
    "name": "mysecgroup",
    "description": "my security group"
  }
}
```

15.15.5. Delete security group

Method	URI	Description
DELETE	/v2.0/security-groups/{security_group_id}	Deletes an OpenStack Networking security group.

This operation deletes an OpenStack Networking security group and its associated security group rules, provided that a port is not associated with the security group.

This operation does not require a request body. This operation does not return a response body.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.15.5.1. Request

This table shows the URI parameters for the delete security group request:

Name	Type	Description
{security_group_id}	UUID	The UUID of the security group.

Example 15.176. Delete security group: JSON request

```
DELETE /v2.0/security-groups/e470bdfc-4869-459b-a561-cb3377efae59
Content-Type: application/json
Accept: application/json
```

This operation does not accept a request body.

15.15.5.2. Response

Example 15.177. Delete security group: JSON response

```
status: 204
```

This operation does not return a response body.

15.16. Security group rules (security-group-rules)

Lists, creates, shows information for, and deletes security group rules.

Method	URI	Description
GET	/v2.0/security-group-rules	Lists a summary of all OpenStack Networking security group rules that the tenant can access.
POST	/v2.0/security-group-rules	Creates an OpenStack Networking security group rule.
GET	/v2.0/security-group-rules/{security-group-rules-id}	Shows detailed information for a security group rule.
DELETE	/v2.0/security-group-rules/{security-group-rules-id}	Deletes a rule from an OpenStack Networking security group.

15.16.1. List security group rules

Method	URI	Description
GET	/v2.0/security-group-rules	Lists a summary of all OpenStack Networking security group rules that the tenant can access.

The list provides the UUID for each security group rule.

Normal response codes: 200

Error response codes: unauthorized (401)

15.16.1.1. Request

Example 15.178. List security group rules: JSON request

```
GET /v2.0/security-group-rules/  
Accept: application/json
```

This operation does not accept a request body.

15.16.1.2. Response

Example 15.179. List security group rules: JSON response

```
{  
  "security_group_rules": [  
    {  
      "direction": "egress",  
      "ethertype": "IPv6",  
      "id": "3c0e45ff-adaf-4124-b083-bf390e5482ff",  
      "port_range_max": null,  
      "port_range_min": null,  
      "protocol": null,  
      "remote_group_id": null,  
      "remote_ip_prefix": null,  
      "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",  
      "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"  
    },  
    {  
      "direction": "egress",  
      "ethertype": "IPv4",  
      "id": "93aa42e5-80db-4581-9391-3a608bd0e448",  
      "port_range_max": null,  
      "port_range_min": null,  
      "protocol": null,  
      "remote_group_id": null,  
      "remote_ip_prefix": null,  
      "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",  
      "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"  
    },  
    {  
      "direction": "ingress",  
      "ethertype": "IPv6",  
      "id": "c0b09f00-1d49-4e64-a0a7-8a186d928138",  

```

```
        "port_range_max": null,  
        "port_range_min": null,  
        "protocol": null,  
        "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",  
        "remote_ip_prefix": null,  
        "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",  
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"  
    },  
    {  
        "direction": "ingress",  
        "ethertype": "IPv4",  
        "id": "f7d45c89-008e-4bab-88ad-d6811724c51c",  
        "port_range_max": null,  
        "port_range_min": null,  
        "protocol": null,  
        "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",  
        "remote_ip_prefix": null,  
        "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",  
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"  
    }  
]  
}
```

15.16.2. Create security group rule

Method	URI	Description
POST	/v2.0/security-group-rules	Creates an OpenStack Networking security group rule.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), conflict (409)

15.16.2.1. Request

Example 15.180. Create security group rule: JSON request

```
{
  "security_group_rule": {
    "direction": "ingress",
    "port_range_min": "80",
    "ethertype": "IPv4",
    "port_range_max": "80",
    "protocol": "tcp",
    "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
    "security_group_id": "a7734e61-b545-452d-a3cd-0189cbd9747a"
  }
}
```

15.16.2.2. Response

Example 15.181. Create security group rule: JSON response

```
{
  "security_group_rule": {
    "direction": "ingress",
    "ethertype": "IPv4",
    "id": "2bc0accf-312e-429a-956e-e4407625eb62",
    "port_range_max": 80,
    "port_range_min": 80,
    "protocol": "tcp",
    "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
    "remote_ip_prefix": null,
    "security_group_id": "a7734e61-b545-452d-a3cd-0189cbd9747a",
    "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
  }
}
```

15.16.3. Show security group rule

Method	URI	Description
GET	/v2.0/security-group-rules/{security-group-rules-id}	Shows detailed information for a security group rule.

The response body contains the following information about the security group rule:

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

15.16.3.1. Request

This table shows the URI parameters for the show security group rule request:

Name	Type	Description
{security_group_rule_id}	UUID	The UUID of the security group rule.

Example 15.182. Show security group rule: JSON request

```
GET /v2.0/security-group-rules/ 3c0e45ff-adaf-4124-b083-bf390e5482ff
Accept: application/json
```

This operation does not accept a request body.

15.16.3.2. Response

Example 15.183. Show security group rule: JSON response

```
{
  "security_group_rule": {
    "direction": "egress",
    "ethertype": "IPv6",
    "id": "3c0e45ff-adaf-4124-b083-bf390e5482ff",
    "port_range_max": null,
    "port_range_min": null,
    "protocol": null,
    "remote_group_id": null,
    "remote_ip_prefix": null,
    "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
    "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
  }
}
```

15.16.4. Delete security group rule

Method	URI	Description
DELETE	/v2.0/security-group-rules/{security-group-rules-id}	Deletes a rule from an OpenStack Networking security group.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404)

15.16.4.1. Request

This table shows the URI parameters for the delete security group rule request:

Name	Type	Description
{security_group_rule_id}	UUID	The UUID of the security group rule.

Example 15.184. Delete security group rule: JSON request

```
DELETE /v2.0/security-group-rules/fc3c327a-b5b5-4cd3-9577-52893289ce08
Content-Type: application/json
Accept: application/json
```

This operation does not accept a request body.

15.16.4.2. Response

Example 15.185. Delete security group rule: JSON response

```
status: 204
```

This operation does not return a response body.

15.17. Subnet pools extension (subnetpools)

Lists, creates, shows details for, updates, and deletes subnet pools.

Method	URI	Description
GET	/v2.0/subnetpools	Lists subnet pools to which the tenant has access.
POST	/v2.0/subnetpools	Creates a subnet pool.
GET	/v2.0/subnetpools/{subnetpool_id}	Shows information for a subnet pool.
PUT	/v2.0/subnetpools/{subnetpool_id}	Updates a subnet pool.
DELETE	/v2.0/subnetpools/{subnetpool_id}	Deletes a subnet pool.

15.17.1. List subnet pools

Method	URI	Description
GET	/v2.0/subnetpools	Lists subnet pools to which the tenant has access.

Default policy settings returns exclusively subnet pools owned by the tenant submitting the request, unless the request is submitted by a user with administrative rights.

Normal response codes: 200

Error response codes: unauthorized (401)

15.17.1.1. Request

This operation does not accept a request body.

15.17.1.2. Response

Example 15.186. List subnet pools: JSON response

```
{
  "subnetpools": [
    {
      "min_prefixlen": "64",
      "address_scope_id": null,
      "default_prefixlen": "64",
      "id": "03f761e6-eee0-43fc-a921-8acf64c14988",
      "max_prefixlen": "64",
      "name": "my-subnet-pool-ipv6",
      "default_quota": null,
      "is_default": false,
      "tenant_id": "9fadcee8aa7c40cdb2114fff7d569c08",
      "prefixes": [
        "2001:db8:0:2::/64",
        "2001:db8::/63"
      ],
      "ip_version": 6,
      "shared": false
    },
    {
      "min_prefixlen": "24",
      "address_scope_id": null,
      "default_prefixlen": "25",
      "id": "f49a1319-423a-4ee6-ba54-1d95a4f6cc68",
      "max_prefixlen": "30",
      "name": "my-subnet-pool-ipv4",
      "default_quota": null,
      "is_default": false,
      "tenant_id": "9fadcee8aa7c40cdb2114fff7d569c08",
      "prefixes": [
        "10.10.0.0/21",
        "192.168.0.0/16"
      ],
      "ip_version": 4,
      "shared": false
    }
  ]
}
```

```
}  
  ]  
}
```

15.17.2. Create subnet pool

Method	URI	Description
POST	/v2.0/subnetpools	Creates a subnet pool.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

15.17.2.1. Request

Example 15.187. Create subnet pool: JSON request

```
{
  "subnetpool": {
    "name": "my-subnet-pool",
    "prefixes": [
      "192.168.0.0/16",
      "10.10.0.0/21"
    ],
    "default_prefixlen": 25,
    "min_prefixlen": 24,
    "max_prefixlen": 30,
    "shared": "false"
  }
}
```

15.17.2.2. Response

Example 15.188. Create subnet pool: JSON response

```
{
  "subnetpool": {
    "min_prefixlen": "24",
    "address_scope_id": null,
    "default_prefixlen": "25",
    "id": "f49a1319-423a-4ee6-ba54-1d95a4f6cc68",
    "max_prefixlen": "30",
    "name": "my-subnet-pool",
    "default_quota": null,
    "is_default": false,
    "tenant_id": "9fadcee8aa7c40cdb2114fff7d569c08",
    "prefixes": [
      "10.10.0.0/21",
      "192.168.0.0/16"
    ],
    "ip_version": 4,
    "shared": false
  }
}
```


15.17.3. Show subnet pool

Method	URI	Description
GET	/v2.0/subnetpools/{subnetpool_id}	Shows information for a subnet pool.

Use the `fields` query parameter to filter the results.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

15.17.3.1. Request

This table shows the URI parameters for the show subnet pool request:

Name	Type	Description
{subnetpool_id}	UUID	The UUID of the subnet pool.

This operation does not accept a request body.

15.17.3.2. Response

Example 15.189. Show subnet pool: JSON response

```
{
  "subnetpool": {
    "min_prefixlen": "64",
    "address_scope_id": null,
    "default_prefixlen": "64",
    "id": "03f761e6-eee0-43fc-a921-8acf64c14988",
    "max_prefixlen": "64",
    "name": "my-subnet-pool",
    "default_quota": null,
    "is_default": false,
    "tenant_id": "9fadcee8aa7c40cdb2114fff7d569c08",
    "created_at": "2016-03-08T20:19:41",
    "prefixes": [
      "2001:db8:0:2::/64",
      "2001:db8::/63"
    ],
    "updated_at": "2016-03-08T20:19:41",
    "ip_version": 6,
    "shared": false
  }
}
```

15.17.4. Update subnet pool

Method	URI	Description
PUT	/v2.0/subnetpools/{subnetpool_id}	Updates a subnet pool.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

15.17.4.1. Request

This table shows the URI parameters for the update subnet pool request:

Name	Type	Description
{subnetpool_id}	UUID	The UUID of the subnet pool.

Example 15.190. Update subnet pool: JSON request

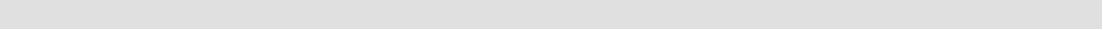
```
{
  "subnetpool": {
    "name": "my-new-subnetpool-name",
    "prefixes": [
      "2001:db8::/64",
      "2001:db8:0:1::/64",
      "2001:db8:0:2::/64"
    ],
    "min_prefixlen": 64,
    "default_prefixlen": 64,
    "max_prefixlen": 64
  }
}
```

15.17.4.2. Response

Example 15.191. Update subnet pool: JSON response

```
{
  "subnetpool": {
    "name": "my-new-subnetpool-name",
    "default_quota": null,
    "is_default": false,
    "tenant_id": "9fadcee8aa7c40cdb2114fff7d569c08",
    "prefixes": [
      "2001:db8::/63",
      "2001:db8:0:2::/64"
    ],
    "min_prefixlen": 64,
    "address_scope_id": null,
    "ip_version": 6,
    "shared": false,
    "default_prefixlen": 64,
    "id": "03f761e6-eee0-43fc-a921-8acf64c14988",
    "max_prefixlen": 64
  }
}
```

```
}
```



15.17.5. Delete subnet pool

Method	URI	Description
DELETE	/v2.0/subnetpools/{subnetpool_id}	Deletes a subnet pool.

The operation fails if any subnets allocated from the subnet pool are still in use.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404)

15.17.5.1. Request

This table shows the URI parameters for the delete subnet pool request:

Name	Type	Description
{subnetpool_id}	UUID	The UUID of the subnet pool.

This operation does not accept a request body.

15.18. Tag extension (tags)

Shows details for, updates, and deletes tags. This extension is available since Mitaka release.

Method	URI	Description
PUT	/v2.0/{resource_type}/{resource_id}/tags	Replaces all tags on the resource.
DELETE	/v2.0/{resource_type}/{resource_id}/tags	Removes all tags on the resource.
GET	/v2.0/{resource_type}/{resource_id}/tags/{tag}	Confirms a given tag is set on the resource.
PUT	/v2.0/{resource_type}/{resource_id}/tags/{tag}	Adds a tag on the resource.
DELETE	/v2.0/{resource_type}/{resource_id}/tags/{tag}	Removes a tag on the resource.

15.18.1. Replace all tags

Method	URI	Description
PUT	/v2.0/{resource_type}/{resource_id}/tags	Replaces all tags on the resource.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), Internal-server-error (500), serviceUnavailable (503)

15.18.1.1. Request

This table shows the URI parameters for the replace all tags request:

Name	Type	Description
{resource_type}	String	The type of resource which the tag is set on.
{resource_id}	UUID	The UUID of resource which the tag is set on.

Example 15.192. Replace all tags: JSON request

```
{
  "tags": [
    "red",
    "blue"
  ]
}
```

15.18.1.2. Response

Example 15.193. Replace all tags: JSON response

```
{
  "tags": [
    "red",
    "blue"
  ]
}
```

15.18.2. Remove all tags

Method	URI	Description
DELETE	/v2.0/{resource_type}/{resource_id}/tags	Removes all tags on the resource.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), Internal-server-error (500), serviceUnavailable (503)

15.18.2.1. Request

This table shows the URI parameters for the remove all tags request:

Name	Type	Description
{resource_type}	String	The type of resource which the tag is set on.
{resource_id}	UUID	The UUID of resource which the tag is set on.

This operation does not accept a request body.

15.18.3. Confirm a tag

Method	URI	Description
GET	/v2.0/{resource_type}/ {resource_id}/tags/{tag}	Confirms a given tag is set on the resource.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), Internal-server-error (500), serviceUnavailable (503)

15.18.3.1. Request

This table shows the URI parameters for the confirm a tag request:

Name	Type	Description
{resource_type}	String	The type of resource which the tag is set on.
{resource_id}	UUID	The UUID of resource which the tag is set on.
{tag}	String	The name for the tag.

This operation does not accept a request body.

15.18.4. Add a tag

Method	URI	Description
PUT	/v2.0/{resource_type}/ {resource_id}/tags/{tag}	Adds a tag on the resource.

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), Internal-server-error (500), serviceUnavailable (503)

15.18.4.1. Request

This table shows the URI parameters for the add a tag request:

Name	Type	Description
{resource_type}	String	The type of resource which the tag is set on.
{resource_id}	UUID	The UUID of resource which the tag is set on.
{tag}	String	The name for the tag.

This operation does not accept a request body.

15.18.5. Remove a tag

Method	URI	Description
DELETE	/v2.0/{resource_type}/{resource_id}/tags/{tag}	Removes a tag on the resource.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), Internal-server-error (500), serviceUnavailable (503)

15.18.5.1. Request

This table shows the URI parameters for the remove a tag request:

Name	Type	Description
{resource_type}	String	The type of resource which the tag is set on.
{resource_id}	UUID	The UUID of resource which the tag is set on.
{tag}	String	The name for the tag.

This operation does not accept a request body.

15.19. VPNaaS 2.0 (CURRENT) (vpn, vpnservices, ikepolicies, ipsecpolicies, endpoint-groups, ipsec-site-connections)

The Virtual-Private-Network-as-a-Service (VPNaaS) extension enables OpenStack tenants to extend private networks across the public telecommunication infrastructure.

This initial implementation of the VPNaaS extension provides:

- Site-to-site VPN that connects two private networks.
- Multiple VPN connections per tenant.
- IKEv1 policy support with 3des, aes-128, aes-256, or aes-192 encryption.
- IPSec policy support with 3des, aes-128, aes-192, or aes-256 encryption, sha1 authentication, ESP, AH, or AH-ESP transform protocol, and tunnel or transport mode encapsulation.
- Dead Peer Detection (DPD) with hold, clear, restart, disabled, or restart-by-peer actions.

This extension introduces these resources:

- **service.** A parent object that associates VPN with a specific subnet and router.
- **ikepolicy.** The Internet Key Exchange (IKE) policy that identifies the authentication and encryption algorithm to use during phase one and two negotiation of a VPN connection.

- **ipsecpolicy**. The IP security policy that specifies the authentication and encryption algorithm and encapsulation mode to use for the established VPN connection.
- **ipsec-site-connection**. Details for the site-to-site IPsec connection, including the peer CIDRs, MTU, authentication mode, peer address, DPD settings, and status.

Method	URI	Description
GET	/v2.0/vpn/vpnservices	Lists all VPN services.
POST	/v2.0/vpn/vpnservices	Creates a VPN service.
GET	/v2.0/vpn/vpnservices/{service_id}	Shows details for a VPN service.
PUT	/v2.0/vpn/vpnservices/{service_id}	Updates a VPN service.
DELETE	/v2.0/vpn/vpnservices/{service_id}	Removes a VPN service.
GET	/v2.0/vpn/ikepolicies	Lists IKE policies.
POST	/v2.0/vpn/ikepolicies	Creates an IKE policy.
GET	/v2.0/vpn/ikepolicies/{ikepolicy_id}	Shows details for an IKE policy.
PUT	/v2.0/vpn/ikepolicies/{ikepolicy_id}	Updates policy settings in an IKE policy.
DELETE	/v2.0/vpn/ikepolicies/{ikepolicy_id}	Removes an IKE policy.
GET	/v2.0/vpn/ipsecpolicies	Lists all IPSec policies.
POST	/v2.0/vpn/ipsecpolicies	Creates an IP security (IPSec) policy.
GET	/v2.0/vpn/ipsecpolicies/{ipsecpolicy_id}	Shows details for an IPSec policy.
PUT	/v2.0/vpn/ipsecpolicies/{ipsecpolicy_id}	Updates policy settings in an IPSec policy.
DELETE	/v2.0/vpn/ipsecpolicies/{ipsecpolicy_id}	Removes an IPSec policy.
GET	/v2.0/vpn/endpoint-groups	Lists VPN endpoint groups.
POST	/v2.0/vpn/endpoint-groups	Creates a VPN endpoint group.
GET	/v2.0/vpn/endpoint-groups/{endpoint_group_id}	Shows details for a VPN endpoint group.
PUT	/v2.0/vpn/endpoint-groups/{endpoint_group_id}	Updates settings for a VPN endpoint group.
DELETE	/v2.0/vpn/endpoint-groups/{endpoint_group_id}	Removes a VPN endpoint group.
GET	/v2.0/vpn/ipsec-site-connections	Lists all IPSec connections.
POST	/v2.0/vpn/ipsec-site-connections	Creates a site-to-site IPSec connection for a service.
GET	/v2.0/vpn/ipsec-site-connections/{connection_id}	Shows details for an IPSec connection.
PUT	/v2.0/vpn/ipsec-site-connections/{connection_id}	Updates connection settings for an IPSec connection.
DELETE	/v2.0/vpn/ipsec-site-connections/{connection_id}	Removes an IPSec connection.

15.19.1. List VPN services

Method	URI	Description
GET	/v2.0/vpn/vpnservices	Lists all VPN services.

The list might be empty.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

15.19.1.1. Request

This operation does not accept a request body.

15.19.1.2. Response

Example 15.194. List VPN services: JSON response

```
{
  "vpnservices": [
    {
      "router_id": "66e3b16c-8ce5-40fb-bb49-ab6d8dc3f2aa",
      "status": "PENDING_CREATE",
      "name": "myservice",
      "external_v6_ip": "2001:db8::1",
      "admin_state_up": true,
      "subnet_id": null,
      "tenant_id": "10039663455a446d8ba2cbb058b0f578",
      "external_v4_ip": "172.32.1.11",
      "id": "5c561d9d-eaea-45f6-ae3e-08d1a7080828",
      "description": ""
    }
  ]
}
```

15.19.2. Create VPN service

Method	URI	Description
POST	/v2.0/vpn/vpnservices	Creates a VPN service.

The service is associated with a router. After you create the service, it can contain multiple VPN connections.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

15.19.2.1. Request

Example 15.195. Create VPN service: JSON request

```
{
  "vpnservice": {
    "subnet_id": null,
    "router_id": "66e3b16c-8ce5-40fb-bb49-ab6d8dc3f2aa",
    "name": "myservice",
    "admin_state_up": true
  }
}
```

15.19.2.2. Response

Example 15.196. Create VPN service: JSON response

```
{
  "vpnservice": {
    "router_id": "66e3b16c-8ce5-40fb-bb49-ab6d8dc3f2aa",
    "status": "PENDING_CREATE",
    "name": "myservice",
    "external_v6_ip": "2001:db8::1",
    "admin_state_up": true,
    "subnet_id": null,
    "tenant_id": "10039663455a446d8ba2cbb058b0f578",
    "external_v4_ip": "172.32.1.11",
    "id": "5c561d9d-eaea-45f6-ae3e-08d1a7080828",
    "description": ""
  }
}
```

15.19.3. Show VPN service details

Method	URI	Description
GET	/v2.0/vpn/vpnservices/{service_id}	Shows details for a VPN service.

If the user is not an administrative user and the VPN service object does not belong to the tenant account for the user, the operation returns the Forbidden (403) response code.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.19.3.1. Request

This table shows the URI parameters for the show vpn service details request:

Name	Type	Description
{service_id}	UUID	The UUID of the VPN service.

This operation does not accept a request body.

15.19.3.2. Response

Example 15.197. Show VPN service details: JSON response

```
{
  "vpnservice": {
    "router_id": "66e3b16c-8ce5-40fb-bb49-ab6d8dc3f2aa",
    "status": "PENDING_CREATE",
    "name": "myservice",
    "external_v6_ip": "2001:db8::1",
    "admin_state_up": true,
    "subnet_id": null,
    "tenant_id": "10039663455a446d8ba2cbb058b0f578",
    "external_v4_ip": "172.32.1.11",
    "id": "5c561d9d-eaea-45f6-ae3e-08d1a7080828",
    "description": ""
  }
}
```

15.19.4. Update VPN service

Method	URI	Description
PUT	/v2.0/vpn/vpnservices/{service_id}	Updates a VPN service.

Updates the attributes of a VPN service. You cannot update a service with a PENDING_* status.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.19.4.1. Request

This table shows the URI parameters for the update vpn service request:

Name	Type	Description
{service_id}	UUID	The UUID of the VPN service.

Example 15.198. Update VPN service: JSON request

```
{
  "vpnservice": {
    "description": "Updated description"
  }
}
```

15.19.4.2. Response

Example 15.199. Update VPN service: JSON response

```
{
  "vpnservice": {
    "router_id": "881b7b30-4efb-407e-a162-5630a7af3595",
    "status": "ACTIVE",
    "name": "myvpn",
    "admin_state_up": true,
    "subnet_id": null,
    "tenant_id": "26de9cd6cae94c8cb9f79d660d628e1f",
    "id": "41bfef97-af4e-4f6b-a5d3-4678859d2485",
    "description": "Updated description"
  }
}
```

15.19.5. Remove VPN service

Method	URI	Description
DELETE	/v2.0/vpn/vpnservices/{service_id}	Removes a VPN service.

If the service has connections, the request is rejected.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.19.5.1. Request

This table shows the URI parameters for the remove vpn service request:

Name	Type	Description
{service_id}	UUID	The UUID of the VPN service.

This operation does not accept a request body.

15.19.6. List IKE policies

Method	URI	Description
GET	/v2.0/vpn/ikepolicies	Lists IKE policies.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

15.19.6.1. Request

This operation does not accept a request body.

15.19.6.2. Response

Example 15.200. List IKE policies: JSON response

```
{
  "ikepolicies": [
    {
      "name": "ikepolicy1",
      "tenant_id": "ccb81365fe36411a9011e90491fe1330",
      "auth_algorithm": "sha1",
      "encryption_algorithm": "aes-256",
      "pfs": "group5",
      "phase1_negotiation_mode": "main",
      "lifetime": {
        "units": "seconds",
        "value": 3600
      },
      "ike_version": "v1",
      "id": "5522aff7-1b3c-48dd-9c3c-b50f016b73db",
      "description": ""
    }
  ]
}
```


15.19.7. Create IKE policy

Method	URI	Description
POST	/v2.0/vpn/ikepolicies	Creates an IKE policy.

The IKE policy is used for phases one and two negotiation of the VPN connection. You can specify both the authentication and encryption algorithms for connections.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

15.19.7.1. Request

Example 15.201. Create IKE policy: JSON request

```
{
  "ikepolicy": {
    "phase1_negotiation_mode": "main",
    "auth_algorithm": "sha1",
    "encryption_algorithm": "aes-128",
    "pfs": "group5",
    "lifetime": {
      "units": "seconds",
      "value": 7200
    },
    "ike_version": "v1",
    "name": "ikepolicy1"
  }
}
```

15.19.7.2. Response

Example 15.202. Create IKE policy: JSON response

```
{
  "ikepolicy": {
    "name": "ikepolicy1",
    "tenant_id": "ccb81365fe36411a9011e90491fe1330",
    "auth_algorithm": "sha1",
    "encryption_algorithm": "aes-128",
    "pfs": "group5",
    "phase1_negotiation_mode": "main",
    "lifetime": {
      "units": "seconds",
      "value": 7200
    },
    "ike_version": "v1",
    "id": "5522aff7-1b3c-48dd-9c3c-b50f016b73db",
    "description": ""
  }
}
```

15.19.8. Show IKE policy details

Method	URI	Description
GET	/v2.0/vpn/ikepolicies/{ikepolicy_id}	Shows details for an IKE policy.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.19.8.1. Request

This table shows the URI parameters for the show ike policy details request:

Name	Type	Description
{ikepolicy_id}	UUID	The UUID of the IKE policy.

This operation does not accept a request body.

15.19.8.2. Response

Example 15.203. Show IKE policy details: JSON response

```
{
  "ikepolicy": {
    "name": "ikepolicy1",
    "tenant_id": "ccb81365fe36411a9011e90491fe1330",
    "auth_algorithm": "sha1",
    "encryption_algorithm": "aes-256",
    "pfs": "group5",
    "phase1_negotiation_mode": "main",
    "lifetime": {
      "units": "seconds",
      "value": 3600
    },
    "ike_version": "v1",
    "id": "5522aff7-1b3c-48dd-9c3c-b50f016b73db",
    "description": ""
  }
}
```

15.19.9. Update IKE policy

Method	URI	Description
PUT	/v2.0/vpn/ikepolicies/{ikepolicy_id}	Updates policy settings in an IKE policy.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.19.9.1. Request

This table shows the URI parameters for the update ike policy request:

Name	Type	Description
{ikepolicy_id}	UUID	The UUID of the IKE policy.

Example 15.204. Update IKE policy: JSON request

```
{
  "ikepolicy": {
    "encryption_algorithm": "aes-256"
  }
}
```

15.19.9.2. Response

Example 15.205. Update IKE policy: JSON response

```
{
  "ikepolicy": {
    "name": "ikepolicy1",
    "tenant_id": "ccb81365fe36411a9011e90491fe1330",
    "auth_algorithm": "sha1",
    "encryption_algorithm": "aes-256",
    "pfs": "group5",
    "phase1_negotiation_mode": "main",
    "lifetime": {
      "units": "seconds",
      "value": 3600
    },
    "ike_version": "v1",
    "id": "5522aff7-1b3c-48dd-9c3c-b50f016b73db",
    "description": ""
  }
}
```

15.19.10. Remove IKE policy

Method	URI	Description
DELETE	/v2.0/vpn/ikepolicies/{ikepolicy_id}	Removes an IKE policy.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.19.10.1. Request

This table shows the URI parameters for the remove ike policy request:

Name	Type	Description
{ikepolicy_id}	UUID	The UUID of the IKE policy.

This operation does not accept a request body.

15.19.11. List IPSec policies

Method	URI	Description
GET	/v2.0/vpn/ipsecpolicies	Lists all IPSec policies.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

15.19.11.1. Request

This operation does not accept a request body.

15.19.11.2. Response

Example 15.206. List IPSec policies: JSON response

```
{
  "ipsecpolicies": [
    {
      "name": "ipsecpolicy1",
      "transform_protocol": "esp",
      "auth_algorithm": "sha1",
      "encapsulation_mode": "tunnel",
      "encryption_algorithm": "aes-128",
      "pfs": "group14",
      "tenant_id": "ccb81365fe36411a9011e90491fe1330",
      "lifetime": {
        "units": "seconds",
        "value": 3600
      },
      "id": "5291b189-fd84-46e5-84bd-78f40c05d69c",
      "description": ""
    }
  ]
}
```

15.19.12. Create IPsec policy

Method	URI	Description
POST	/v2.0/vpn/ipsecpolicies	Creates an IP security (IPsec) policy.

The IPsec policy specifies the authentication and encryption algorithms and encapsulation mode to use for the established VPN connection.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

15.19.12.1. Request

Example 15.207. Create IPsec policy: JSON request

```
{
  "ipsecpolicy": {
    "name": "ipsecpolicy1",
    "transform_protocol": "esp",
    "auth_algorithm": "sha1",
    "encapsulation_mode": "tunnel",
    "encryption_algorithm": "aes-128",
    "pfs": "group5",
    "lifetime": {
      "units": "seconds",
      "value": 7200
    }
  }
}
```

15.19.12.2. Response

Example 15.208. Create IPsec policy: JSON response

```
{
  "ipsecpolicy": {
    "name": "ipsecpolicy1",
    "transform_protocol": "esp",
    "auth_algorithm": "sha1",
    "encapsulation_mode": "tunnel",
    "encryption_algorithm": "aes-128",
    "pfs": "group5",
    "tenant_id": "ccb81365fe36411a9011e90491fe1330",
    "lifetime": {
      "units": "seconds",
      "value": 7200
    },
    "id": "5291b189-fd84-46e5-84bd-78f40c05d69c",
    "description": ""
  }
}
```

15.19.13. Show IPsec policy

Method	URI	Description
GET	/v2.0/vpn/ipsecpolicies/{ipsecpolicy_id}	Shows details for an IPsec policy.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.19.13.1. Request

This table shows the URI parameters for the show ipsec policy request:

Name	Type	Description
{ipsecpolicy_id}	UUID	The UUID of the IPsec policy.

This operation does not accept a request body.

15.19.13.2. Response

Example 15.209. Show IPsec policy: JSON response

```
{
  "ipsecpolicy": {
    "name": "ipsecpolicy1",
    "transform_protocol": "esp",
    "auth_algorithm": "sha1",
    "encapsulation_mode": "tunnel",
    "encryption_algorithm": "aes-128",
    "pfs": "group14",
    "tenant_id": "ccb81365fe36411a9011e90491fe1330",
    "lifetime": {
      "units": "seconds",
      "value": 3600
    },
    "id": "5291b189-fd84-46e5-84bd-78f40c05d69c",
    "description": ""
  }
}
```

15.19.14. Update IPsec policy

Method	URI	Description
PUT	/v2.0/vpn/ipsecpolicies/{ipsecpolicy_id}	Updates policy settings in an IPsec policy.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.19.14.1. Request

This table shows the URI parameters for the update ipsec policy request:

Name	Type	Description
{ipsecpolicy_id}	UUID	The UUID of the IPsec policy.

Example 15.210. Update IPsec policy: JSON request

```
{
  "ipsecpolicy": {
    "pfs": "group14"
  }
}
```

15.19.14.2. Response

Example 15.211. Update IPsec policy: JSON response

```
{
  "ipsecpolicy": {
    "name": "ipsecpolicy1",
    "transform_protocol": "esp",
    "auth_algorithm": "sha1",
    "encapsulation_mode": "tunnel",
    "encryption_algorithm": "aes-128",
    "pfs": "group14",
    "tenant_id": "ccb81365fe36411a9011e90491fe1330",
    "lifetime": {
      "units": "seconds",
      "value": 3600
    },
    "id": "5291b189-fd84-46e5-84bd-78f40c05d69c",
    "description": ""
  }
}
```


15.19.15. Remove IPSec policy

Method	URI	Description
DELETE	/v2.0/vpn/ipsecpolicies/{ipsecpolicy_id}	Removes an IPSec policy.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.19.15.1. Request

This table shows the URI parameters for the remove ipsec policy request:

Name	Type	Description
{ipsecpolicy_id}	UUID	The UUID of the IPSec policy.

This operation does not accept a request body.

15.19.16. List VPN endpoint groups

Method	URI	Description
GET	/v2.0/vpn/endpoint-groups	Lists VPN endpoint groups.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

15.19.16.1. Request

This operation does not accept a request body.

15.19.16.2. Response

Example 15.212. List VPN endpoint groups: JSON response

```
{
  "endpoint_groups": [
    {
      "description": "",
      "tenant_id": "4ad57e7ce0b24fca8f12b9834d91079d",
      "endpoints": [
        "a3da778c-adfb-46db-88b3-d2ce53290a89"
      ],
      "type": "subnet",
      "id": "6bf34c7c-864c-4948-a6d4-db791669f9d4",
      "name": "locals"
    },
    {
      "description": "",
      "tenant_id": "4ad57e7ce0b24fca8f12b9834d91079d",
      "endpoints": [
        "10.2.0.0/24",
        "10.3.0.0/24"
      ],
      "type": "cidr",
      "id": "6ecd9cf3-ca64-46c7-863f-f2eb1b9e838a",
      "name": "peers"
    }
  ]
}
```

15.19.17. Create VPN endpoint group

Method	URI	Description
POST	/v2.0/vpn/endpoint-groups	Creates a VPN endpoint group.

The endpoint group contains one or more endpoints of a specific type that you can use to create a VPN connections.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

15.19.17.1. Request

Example 15.213. Create VPN endpoint group: JSON request

```
{
  "endpoint_group": {
    "endpoints": [
      "10.2.0.0/24",
      "10.3.0.0/24"
    ],
    "type": "cidr",
    "name": "peers"
  }
}
```

15.19.17.2. Response

Example 15.214. Create VPN endpoint group: JSON response

```
{
  "endpoint_group": {
    "description": "",
    "tenant_id": "4ad57e7ce0b24fca8f12b9834d91079d",
    "endpoints": [
      "10.2.0.0/24",
      "10.3.0.0/24"
    ],
    "type": "cidr",
    "id": "6ecd9cf3-ca64-46c7-863f-f2eb1b9e838a",
    "name": "peers"
  }
}
```

15.19.18. Show VPN endpoint group

Method	URI	Description
GET	/v2.0/vpn/endpoint-groups/{endpoint_group_id}	Shows details for a VPN endpoint group.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.19.18.1. Request

This table shows the URI parameters for the show vpn endpoint group request:

Name	Type	Description
{endpoint_group_id}	UUID	The UUID of the VPN endpoint group.

This operation does not accept a request body.

15.19.18.2. Response

Example 15.215. Show VPN endpoint group: JSON response

```
{
  "endpoint_group": {
    "description": "",
    "tenant_id": "4ad57e7ce0b24fca8f12b9834d91079d",
    "endpoints": [
      "10.2.0.0/24",
      "10.3.0.0/24"
    ],
    "type": "cidr",
    "id": "6ecd9cf3-ca64-46c7-863f-f2eb1b9e838a",
    "name": "peers"
  }
}
```

15.19.19. Update VPN endpoint group

Method	URI	Description
PUT	/v2.0/vpn/endpoint-groups/{endpoint_group_id}	Updates settings for a VPN endpoint group.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.19.19.1. Request

This table shows the URI parameters for the update vpn endpoint group request:

Name	Type	Description
{endpoint_group_id}	UUID	The UUID of the VPN endpoint group.

Example 15.216. Update VPN endpoint group: JSON request

```
{
  "endpoint_group": {
    "description": "New description"
  }
}
```

15.19.19.2. Response

Example 15.217. Update VPN endpoint group: JSON response

```
{
  "endpoint_group": {
    "description": "New description",
    "tenant_id": "4ad57e7ce0b24fca8f12b9834d91079d",
    "endpoints": [
      "10.2.0.0/24",
      "10.3.0.0/24"
    ],
    "type": "cidr",
    "id": "6ecd9cf3-ca64-46c7-863f-f2eb1b9e838a",
    "name": "peers"
  }
}
```

15.19.20. Remove VPN endpoint group

Method	URI	Description
DELETE	/v2.0/vpn/endpoint-groups/{endpoint_group_id}	Removes a VPN endpoint group.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.19.20.1. Request

This table shows the URI parameters for the remove vpn endpoint group request:

Name	Type	Description
{endpoint_group_id}	UUID	The UUID of the VPN endpoint group.

This operation does not accept a request body.

15.19.21. List IPSec connections

Method	URI	Description
GET	/v2.0/vpn/ipsec-site-connections	Lists all IPSec connections.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

15.19.21.1. Request

This operation does not accept a request body.

15.19.21.2. Response

Example 15.218. List IPSec connections: JSON response

```
{
  "ipsec_site_connections": [
    {
      "status": "PENDING CREATE",
      "psk": "secret",
      "initiator": "bi-directional",
      "name": "vpnconnection1",
      "admin_state_up": true,
      "tenant_id": "10039663455a446d8ba2cbb058b0f578",
      "auth_mode": "psk",
      "peer_cidrs": [],
      "mtu": 1500,
      "peer_ep_group_id": "9ad5a7e0-6dac-41b4-b20d-a7b8645fddf1",
      "ikepolicy_id": "9b00d6b0-6c93-4ca5-9747-b8ade7bb514f",
      "vpnservice_id": "5c561d9d-eaea-45f6-ae3e-08d1a7080828",
      "dpd": {
        "action": "hold",
        "interval": 30,
        "timeout": 120
      },
      "route_mode": "static",
      "ipsecpolicy_id": "e6e23d0c-9519-4d52-8ea4-5b1f96d857b1",
      "local_ep_group_id": "3e1815dd-e212-43d0-8f13-b494fa553e68",
      "peer_address": "172.24.4.226",
      "peer_id": "172.24.4.226",
      "id": "851f280f-5639-4ea3-81aa-e298525ab74b",
      "description": ""
    }
  ]
}
```

15.19.22. Create IPsec connection

Method	URI	Description
POST	/v2.0/vpn/ipsec-site-connections	Creates a site-to-site IPsec connection for a service.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

15.19.22.1. Request

Example 15.219. Create IPsec connection: JSON request

```
{
  "ipsec_site_connection": {
    "psk": "secret",
    "initiator": "bi-directional",
    "ipsecpolicy_id": "e6e23d0c-9519-4d52-8ea4-5b1f96d857b1",
    "admin_state_up": true,
    "mtu": "1500",
    "peer_ep_group_id": "9ad5a7e0-6dac-41b4-b20d-a7b8645fddf1",
    "ikepolicy_id": "9b00d6b0-6c93-4ca5-9747-b8ade7bb514f",
    "vpnservice_id": "5c561d9d-eaea-45f6-ae3e-08d1a7080828",
    "local_ep_group_id": "3e1815dd-e212-43d0-8f13-b494fa553e68",
    "peer_address": "172.24.4.233",
    "peer_id": "172.24.4.233",
    "name": "vpnconnection1"
  }
}
```

15.19.22.2. Response

Example 15.220. Create IPsec connection: JSON response

```
{
  "ipsec_site_connection": {
    "status": "PENDING_CREATE",
    "psk": "secret",
    "initiator": "bi-directional",
    "name": "vpnconnection1",
    "admin_state_up": true,
    "tenant_id": "10039663455a446d8ba2cbb058b0f578",
    "auth_mode": "psk",
    "peer_cidrs": [],
    "mtu": 1500,
    "peer_ep_group_id": "9ad5a7e0-6dac-41b4-b20d-a7b8645fddf1",
    "ikepolicy_id": "9b00d6b0-6c93-4ca5-9747-b8ade7bb514f",
    "vpnservice_id": "5c561d9d-eaea-45f6-ae3e-08d1a7080828",
    "dpd": {
      "action": "hold",
      "interval": 30,
      "timeout": 120
    },
    "route_mode": "static",
    "ipsecpolicy_id": "e6e23d0c-9519-4d52-8ea4-5b1f96d857b1",
    "local_ep_group_id": "3e1815dd-e212-43d0-8f13-b494fa553e68",
  }
}
```



```
    "peer_address": "172.24.4.233",  
    "peer_id": "172.24.4.233",  
    "id": "851f280f-5639-4ea3-81aa-e298525ab74b",  
    "description": ""  
  }  
}
```

15.19.23. Show IPsec connection

Method	URI	Description
GET	/v2.0/vpn/ipsec-site-connections/{connection_id}	Shows details for an IPsec connection.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

15.19.23.1. Request

This table shows the URI parameters for the show ipsec connection request:

Name	Type	Description
{connection_id}	UUID	The UUID of the IPsec site-to-site connection.

This operation does not accept a request body.

15.19.23.2. Response

Example 15.221. Show IPsec connection: JSON response

```
{
  "ipsec_site_connection": {
    "status": "DOWN",
    "psk": "secret",
    "initiator": "bi-directional",
    "name": "vpnconnection1",
    "admin_state_up": true,
    "tenant_id": "10039663455a446d8ba2cbb058b0f578",
    "auth_mode": "psk",
    "peer_cidrs": [],
    "mtu": 1500,
    "peer_ep_group_id": "9ad5a7e0-6dac-41b4-b20d-a7b8645fddf1",
    "ikepolicy_id": "9b00d6b0-6c93-4ca5-9747-b8ade7bb514f",
    "vpnservice_id": "5c561d9d-eaea-45f6-ae3e-08d1a7080828",
    "dpd": {
      "action": "hold",
      "interval": 30,
      "timeout": 120
    },
    "route_mode": "static",
    "ipsecpolicy_id": "e6e23d0c-9519-4d52-8ea4-5b1f96d857b1",
    "local_ep_group_id": "3e1815dd-e212-43d0-8f13-b494fa553e68",
    "peer_address": "172.24.4.226",
    "peer_id": "172.24.4.226",
    "id": "851f280f-5639-4ea3-81aa-e298525ab74b",
    "description": ""
  }
}
```

15.19.24. Update IPsec connection

Method	URI	Description
PUT	/v2.0/vpn/ipsec-site-connections/{connection_id}	Updates connection settings for an IPsec connection.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

15.19.24.1. Request

This table shows the URI parameters for the update ipsec connection request:

Name	Type	Description
{connection_id}	UUID	The UUID of the IPsec site-to-site connection.

Example 15.222. Update IPsec connection: JSON request

```
{
  "ipsec_site_connection": {
    "mtu": "2000"
  }
}
```

15.19.24.2. Response

Example 15.223. Update IPsec connection: JSON response

```
{
  "ipsec_site_connection": {
    "status": "DOWN",
    "psk": "secret",
    "initiator": "bi-directional",
    "name": "vpnconnection1",
    "admin_state_up": true,
    "tenant_id": "10039663455a446d8ba2cbb058b0f578",
    "auth_mode": "psk",
    "peer_cidrs": [],
    "mtu": 2000,
    "peer_ep_group_id": "9ad5a7e0-6dac-41b4-b20d-a7b8645fddf1",
    "ikepolicy_id": "9b00d6b0-6c93-4ca5-9747-b8ade7bb514f",
    "vpnservice_id": "5c561d9d-eaea-45f6-ae3e-08d1a7080828",
    "dpd": {
      "action": "hold",
      "interval": 30,
      "timeout": 120
    },
    "route_mode": "static",
    "ipsecpolicy_id": "e6e23d0c-9519-4d52-8ea4-5b1f96d857b1",
    "local_ep_group_id": "3e1815dd-e212-43d0-8f13-b494fa553e68",
    "peer_address": "172.24.4.233",
    "peer_id": "172.24.4.233",
    "id": "851f280f-5639-4ea3-81aa-e298525ab74b",
    "description": "New description"
  }
}
```

```
}  
}
```

15.19.25. Remove IPsec connection

Method	URI	Description
DELETE	/v2.0/vpn/ipsec-site-connections/{connection_id}	Removes an IPsec connection.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

15.19.25.1. Request

This table shows the URI parameters for the remove ipsec connection request:

Name	Type	Description
{connection_id}	UUID	The UUID of the IPsec site-to-site connection.

This operation does not accept a request body.

15.20. Network IP availability and usage stats

List and show the network IP usage stats of all networks and a specified network. These operations are available in the Mitaka release (April 2016).

Method	URI	Description
GET	/v2.0/network-ip-availabilities	List network IP availability of all networks.
GET	/v2.0/network-ip-availabilities	Show network IP availability of all networks.

15.20.1. Network IP Availability

Method	URI	Description
GET	/v2.0/network-ip-availabilities	List network IP availability of all networks.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

15.20.1.1. Request

This operation does not accept a request body.

15.20.1.2. Response

Example 15.224. Network IP Availability: JSON response

```
{
  "network_ip_availabilities": [
    {
      "id": "6ed07354-852f-41f8-8555-511d995cd17e",
      "name": "net3",
      "subnet_ip_availability": [
        {
          "cidr": "40.0.0.0/24",
          "ip_version": 4,
          "name": "",
          "subnet_id": "6df63baf-7c98-4cf6-a9a3-f0b5cbceaaa0",
          "total_ips": 253,
          "used_ips": 2
        }
      ],
      "tenant_id": "test-tenant",
      "total_ips": 253,
      "used_ips": 2
    },
    {
      "id": "aba3b29b-c119-4b45-afbd-88e500acd970",
      "name": "net1",
      "subnet_ip_availability": [
        {
          "cidr": "10.0.0.0/24",
          "ip_version": 4,
          "name": "",
          "subnet_id": "df3f70c7-8096-47b6-95ae-b3a7d62b937f",
          "total_ips": 253,
          "used_ips": 3
        }
      ],
      "tenant_id": "test-tenant",
      "total_ips": 253,
      "used_ips": 3
    },
    {
      "id": "cf98f2d0-67db-4ed8-a9d5-06033338dc1b",
      "name": "net2",
```

```
        "subnet_ip_availability": [],  
        "tenant_id": "test-tenant",  
        "total_ips": 0,  
        "used_ips": 0  
    }  
]  
}
```

15.20.2. Network IP Availability

Method	URI	Description
GET	/v2.0/network-ip-availabilities	Show network IP availability of all networks.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

15.20.2.1. Request

This operation does not accept a request body.

15.20.2.2. Response

Example 15.225. Network IP Availability: JSON response

```
{
  "network_ip_availability": {
    "id": "aba3b29b-c119-4b45-afbd-88e500acd970",
    "name": "net1",
    "subnet_ip_availability": [
      {
        "cidr": "10.0.0.0/24",
        "ip_version": 4,
        "name": "",
        "subnet_id": "df3f70c7-8096-47b6-95ae-b3a7d62b937f",
        "total_ips": 253,
        "used_ips": 3
      }
    ],
    "tenant_id": "test-tenant",
    "total_ips": 253,
    "used_ips": 3
  }
}
```


16. Object Storage API v1 (SUPPORTED)

Manages the accounts, containers, and objects in the Object Storage system.

To run the cURL command examples for these requests, set these environment variables:

- `publicURL`. The public URL that is the HTTP endpoint from where you can access Object Storage. It includes the Object Storage API version number and your account name. For example, `https://23.253.72.207/v1/my_account`.
- `token`. The authentication token for Object Storage.

To obtain these values, run the **swift stat -v** command.

As shown in this example, the public URL appears in the `StorageURL` field, and the token appears in the `Auth Token` field:

```
StorageURL: https://23.253.72.207/v1/my_account
Auth Token: {token}
Account: my_account
Containers: 2
Objects: 3
Bytes: 47
Meta Book: MobyDick
X-Timestamp: 1389453423.35964
X-Trans-Id: txee55498935404a2caad89-0052dd3b77
Content-Type: text/plain; charset=utf-8
Accept-Ranges: bytes
```

For a complete description of HTTP 1.1 header definitions, see [Header Field Definitions](#).

The API returns the `Range Not Satisfiable (416)` response code for any ranged **GET** requests that specify more than:

- Fifty ranges.
- Three overlapping ranges.
- Eight non-increasing ranges.

Method	URI	Description
Discoverability		
GET	<code>/info{?swiftinfo_sig, swiftinfo_expires}</code>	Lists the activated capabilities for this version of the OpenStack Object Storage API.
Endpoints		
GET	<code>/v1/endpoints</code>	Lists endpoints for an object, account, or container.
Accounts		
GET	<code>/v1/{account}{?limit,marker, end_marker, format, prefix, delimiter}</code>	Shows details for an account and lists containers, sorted by name, in the account.
POST	<code>/v1/{account}</code>	Creates, updates, or deletes account metadata.
HEAD	<code>/v1/{account}</code>	Shows metadata for an account.
Containers		

Method	URI	Description
GET	/v1/{account}/{container}{?limit, marker, end_marker, prefix, format, delimiter, path}	Shows details for a container and lists objects, sorted by name, in the container.
PUT	/v1/{account}/{container}	Creates a container.
DELETE	/v1/{account}/{container}	Deletes an empty container.
POST	/v1/{account}/{container}	Creates, updates, or deletes custom metadata for a container.
HEAD	/v1/{account}/{container}	Shows container metadata, including the number of objects and the total bytes of all objects stored in the container.
Objects		
GET	/v1/{account}/{container}/{object}{?temp_url_sig, temp_url_expires, filename, multipart-manifest}	Downloads the object content and gets the object metadata.
PUT	/v1/{account}/{container}/{object}{?multipart-manifest, temp_url_sig, temp_url_expires, filename}	Creates an object with data content and metadata, or replaces an existing object with data content and metadata.
COPY	/v1/{account}/{container}/{object}	Copies an object to another object in the object store.
DELETE	/v1/{account}/{container}/{object}{?multipart-manifest}	Permanently deletes an object from the object store.
HEAD	/v1/{account}/{container}/{object}{?temp_url_sig, temp_url_expires, filename}	Shows object metadata.
POST	/v1/{account}/{container}/{object}	Creates or updates object metadata.

16.1. Discoverability

If configured, lists the activated capabilities for this version of the OpenStack Object Storage API.

Method	URI	Description
GET	/info{?swiftinfo_sig, swiftinfo_expires}	Lists the activated capabilities for this version of the OpenStack Object Storage API.

16.1.1. List activated capabilities

Method	URI	Description
GET	/info{?swiftinfo_sig, swiftinfo_expires}	Lists the activated capabilities for this version of the OpenStack Object Storage API.

Normal response codes: 200

16.1.1.1. Request

This table shows the query parameters for the list activated capabilities request:

Name	Type	Description
swiftinfo_sig	Char (Optional)	A hash-based message authentication code (HMAC) that enables access to administrator-only information. To use this parameter, the swiftinfo_expires parameter is also required.
swiftinfo_expires	Int (Optional)	Filters the response by the expiration date and time in UNIX Epoch time stamp format . For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.

This operation does not accept a request body.

16.1.1.2. Response

Example 16.1. List activated capabilities: JSON response

```
{
  "swift": {
    "version": "1.11.0"
  },
  "staticweb": {},
  "tempurl": {}
}
```

16.2. Endpoints

If configured, lists endpoints for an account.

Method	URI	Description
GET	/v1/endpoints	Lists endpoints for an object, account, or container.

16.2.1. List endpoints

Method	URI	Description
GET	/v1/endpoints	Lists endpoints for an object, account, or container.

When the cloud provider enables middleware to list the `/endpoints/` path, software that needs data location information can use this call to avoid network overhead. The cloud provider can map the `/endpoints/` path to another resource, so this exact resource might vary from provider to provider. Because it goes straight to the middleware, the call is not authenticated, so be sure you have tightly secured the environment and network when using this call.

Normal response codes: 201

16.2.1.1. Request

This operation does not accept a request body.

16.2.1.2. Response

Example 16.2. List endpoints: JSON response

```
{
  "endpoints": [
    "http://storage02.swiftdrive:6002/d2/617/AUTH_dev",
    "http://storage01.swiftdrive:6002/d8/617/AUTH_dev",
    "http://storage01.swiftdrive:6002/d11/617/AUTH_dev"
  ],
  "headers": {}
}
```

Example 16.3. List endpoints: JSON response

```
{
  "endpoints": [
    "http://storage01.swiftdrive.com:6008/d8/583/AUTH_dev/EC_cont1/obj",
    "http://storage02.swiftdrive.com:6008/d2/583/AUTH_dev/EC_cont1/obj",
    "http://storage02.swiftdrive.com:6006/d3/583/AUTH_dev/EC_cont1/obj",
    "http://storage02.swiftdrive.com:6008/d5/583/AUTH_dev/EC_cont1/obj",
    "http://storage01.swiftdrive.com:6007/d7/583/AUTH_dev/EC_cont1/obj",
    "http://storage02.swiftdrive.com:6007/d4/583/AUTH_dev/EC_cont1/obj",
    "http://storage01.swiftdrive.com:6006/d6/583/AUTH_dev/EC_cont1/obj"
  ],
  "headers": {
    "X-Backend-Storage-Policy-Index": "2"
  }
}
```

16.3. Accounts

Lists containers for an account. Creates, updates, shows, and deletes account metadata.

Account metadata operations work differently than container and object metadata operations work. Depending on the contents of your **POST** account metadata request, the Object Storage API updates the metadata in one of these ways:

Table 16.1. Account metadata operations

POST request body contains	Description
A metadata key without a value. The metadata key already exists for the account.	The API removes the metadata item from the account.
A metadata key without a value. The metadata key does not already exist for the account.	The API ignores the metadata key.
A metadata key value. The metadata key already exists for the account.	The API updates the metadata key value for the account.
A metadata key value. The metadata key does not already exist for the account.	The API adds the metadata key and value pair, or item, to the account.
One or more account metadata items are omitted. The metadata items already exist for the account.	The API does not change the existing metadata items.

For these requests, specifying the `X-Remove-Account-Meta-*` request header for the key with any value is equivalent to specifying the `X-Account-Meta-*` request header with an empty value.

Metadata keys must be treated as case-insensitive at all times. These keys can contain ASCII 7-bit characters that are not control (0-31) characters, DEL, or a separator character, according to [HTTP/1.1](#). Also, Object Storage does not support the underscore character, which it silently converts to a hyphen.

The metadata values in Object Storage do not follow HTTP/1.1 rules for character encodings. You must use a UTF-8 encoding to get a byte array for any string that contains characters that are not in the 7-bit ASCII 0-127 range. Otherwise, Object Storage returns the 404 response code for ISO-8859-1 characters in the 128-255 range, which is a direct violation of the HTTP/1.1 [basic rules](#).

Method	URI	Description
GET	<code>/v1/{account}{?limit,marker,end_marker,format,prefix,delimiter}</code>	Shows details for an account and lists containers, sorted by name, in the account.
POST	<code>/v1/{account}</code>	Creates, updates, or deletes account metadata.
HEAD	<code>/v1/{account}</code>	Shows metadata for an account.

16.3.1. Show account details and list containers

Method	URI	Description
GET	/v1/{account}{?limit,marker,end_marker,format,prefix,delimiter}	Shows details for an account and lists containers, sorted by name, in the account.

The sort order for the name is based on a binary comparison, a single built-in collating sequence that compares string data by using the SQLite `memcmp()` function, regardless of text encoding. See [Collating Sequences](#).

Example requests and responses:

- Show account details and list containers and ask for a JSON response:

```
curl -i $publicURL?format=json -X GET -H "X-Auth-Token: $token"
```

- List containers and ask for an XML response:

```
curl -i $publicURL?format=xml -X GET -H "X-Auth-Token: $token"
```

The response body returns a list of containers. The default response (`text/plain`) returns one container per line.

If you use query parameters to page through a long list of containers, you have reached the end of the list if the number of items in the returned list is less than the request `limit` value. The list contains more items if the number of items in the returned list equals the `limit` value.

When asking for a list of containers and there are none, the response behavior changes depending on whether the request format is text, JSON, or XML. For a text response, you get a 204, because there is no content. However, for a JSON or XML response, you get a 200 with content indicating an empty array.

If the request succeeds, the operation returns one of these status codes:

- OK (200). Success. The response body lists the containers.
- No Content (204). Success. The response body shows no containers. Either the account has no containers or you are paging through a long list of names by using the `marker`, `limit`, or `end_marker` query parameter and you have reached the end of the list.

Normal response codes: 200204

16.3.1.1. Request

This table shows the header parameters for the show account details and list containers request:

Name	Type	Description
X-Auth-Token	String	Authentication token.

Name	Type	Description
	<i>(Required)</i>	
X-Newest	Boolean <i>(Optional)</i>	If set to <code>true</code> , Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to <code>true</code> is more expensive for the back end, use it only when it is absolutely needed.
Accept	String <i>(Optional)</i>	Instead of using the <code>format</code> query parameter, set this header to <code>application/json</code> , <code>application/xml</code> , or <code>text/xml</code> .
X-Trans-Id-Extra	UUID <i>(Optional)</i>	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the show account details and list containers request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.

This table shows the query parameters for the show account details and list containers request:

Name	Type	Description
limit	Int <i>(Optional)</i>	For an integer value <i>n</i> , limits the number of results to <i>n</i> .
marker	String <i>(Optional)</i>	For a string value, <i>x</i> , returns container names that are greater than the marker value.
end_marker	String <i>(Optional)</i>	For a string value, <i>x</i> , returns container names that are less than the marker value.
format	String <i>(Optional)</i>	The response format. Valid values are <code>json</code>, <code>xml</code>, or <code>plain</code>. The default is <code>plain</code>. If you append the <code>format=xml</code> or <code>format=json</code> query parameter to the storage account URL, the response shows extended container information serialized in that format. If you append the <code>format=plain</code> query parameter, the response lists the container names separated by newlines.
prefix	String <i>(Optional)</i>	Prefix value. Named items in the response begin with this value.

Name	Type	Description
delimiter	Char (Optional)	Delimiter value, which returns the object names that are nested in the container. If you do not set a prefix and set the delimiter to "/" you may get unexpected results where all the objects are returned instead of only those with the delimiter set.

This operation does not accept a request body.

16.3.1.2. Response

Example 16.4. List containers response: HTTP and JSON

```
HTTP/1.1 200 OK
Content-Length: 96
X-Account-Object-Count: 1
X-Timestamp: 1389453423.35964
X-Account-Meta-Subject: Literature
X-Account-Bytes-Used: 14
X-Account-Container-Count: 2
Content-Type: application/json; charset=utf-8
Accept-Ranges: bytes
X-Trans-Id: tx274a77a8975c4a66aeb24-0052d95365
Date: Fri, 17 Jan 2014 15:59:33 GMT
```

```
[
  {
    "count": 0,
    "bytes": 0,
    "name": "janeaugsten"
  },
  {
    "count": 1,
    "bytes": 14,
    "name": "marktwain"
  }
]
```

This table shows the body parameters for the show account details and list containers response:

Name	Type	Description
name	String (Required)	The name of the container.
count	Int (Required)	The number of objects in the container.
bytes	Int (Required)	The total number of bytes that are stored in Object Storage for the account.

Example 16.5. List containers response: HTTP and XML

```
HTTP/1.1 200 OK
Content-Length: 262
X-Account-Object-Count: 1
X-Timestamp: 1389453423.35964
X-Account-Meta-Subject: Literature
```



```
X-Account-Bytes-Used: 14
X-Account-Container-Count: 2
Content-Type: application/xml; charset=utf-8
Accept-Ranges: bytes
X-Trans-Id: tx69f60bc9f7634a01988e6-0052d9544b
Date: Fri, 17 Jan 2014 16:03:23 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<account name="my_account">
  <container>
    <name>janeausten</name>
    <count>0</count>
    <bytes>0</bytes>
  </container>
  <container>
    <name>marktwain</name>
    <count>1</count>
    <bytes>14</bytes>
  </container>
</account>
```

This operation does not return a response body.

16.3.2. Create, update, or delete account metadata

Method	URI	Description
POST	/v1/{account}	Creates, updates, or deletes account metadata.

To create, update, or delete metadata, use the X-Account-Meta-{name} request header, where {name} is the name of the metadata item.

Subsequent requests for the same key and value pair overwrite the existing value.

To delete a metadata header, send an empty value for that header, such as for the X-Account-Meta-Book header. If the tool you use to communicate with Object Storage, such as an older version of cURL, does not support empty headers, send the X-Remove-Account-Meta-{name} header with an arbitrary value. For example, X-Remove-Account-Meta-Book: x. The operation ignores the arbitrary value.

If the container already has other custom metadata items, a request to create, update, or delete metadata does not affect those items.

This operation does not accept a request body.

Example requests and responses:

- Create account metadata:

```
curl -i $publicURL -X POST -H "X-Auth-Token: $token" -H "X-Account-Meta-Book: MobyDick" -H "X-Account-Meta-Subject: Literature"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx8c2dd6aee35442a4a5646-0052d954fb
Date: Fri, 17 Jan 2014 16:06:19 GMT
```

- Update account metadata:

```
curl -i $publicURL -X POST -H "X-Auth-Token: $token" -H "X-Account-Meta-Subject: AmericanLiterature"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx1439b96137364ab581156-0052d95532
Date: Fri, 17 Jan 2014 16:07:14 GMT
```

- Delete account metadata:

```
curl -i $publicURL -X POST -H "X-Auth-Token: $token" -H "X-Remove-Account-Meta-Subject: x"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
```

```
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx411cf57701424da99948a-0052d9556f
Date: Fri, 17 Jan 2014 16:08:15 GMT
```

If the request succeeds, the operation returns the No Content (204) response code.

To confirm your changes, issue a show account metadata request.

Normal response codes: 204

16.3.2.1. Request

This table shows the header parameters for the create, update, or delete account metadata request:

Name	Type	Description
X-Auth-Token	String (Required)	Authentication token.
X-Account-Meta-Temp-URL-Key	String (Optional)	The secret key value for temporary URLs.
X-Account-Meta-Temp-URL-Key-2	String (Optional)	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.
X-Account-Meta-name	String (Optional)	The account metadata. The {name} is the name of metadata item that you want to add, update, or delete. To delete this item, send an empty value in this header. You must specify an X-Account-Meta-{name} header for each metadata item (for each {name}) that you want to add, update, or delete.
Content-Type	String (Optional)	Changes the MIME type for the object.
X-Detect-Content-Type	Boolean (Optional)	If set to true, Object Storage guesses the content type based on the file extension and ignores the value sent in the Content-Type header, if present.
X-Trans-Id-Extra	UUID (Optional)	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the create, update, or delete account metadata request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.

This operation does not accept a request body.

16.3.2.2. Response

This operation does not return a response body.

16.3.3. Show account metadata

Method	URI	Description
HEAD	/v1/{account}	Shows metadata for an account.

Metadata for the account includes:

- Number of containers
- Number of objects
- Total number of bytes that are stored in Object Storage for the account

Because the storage system can store large amounts of data, take care when you represent the total bytes response as an integer; when possible, convert it to a 64-bit unsigned integer if your platform supports that primitive type.

Do not include metadata headers in this request.

Show account metadata request:

```
curl -i $publicURL -X HEAD -H "X-Auth-Token: $token"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
X-Account-Object-Count: 1
X-Account-Meta-Book: MobyDick
X-Timestamp: 1389453423.35964
X-Account-Bytes-Used: 14
X-Account-Container-Count: 2
Content-Type: text/plain; charset=utf-8
Accept-Ranges: bytes
X-Trans-Id: txafb3504870144b8ca40f7-0052d955d4
Date: Fri, 17 Jan 2014 16:09:56 GMT
```

If the account or authentication token is not valid, the operation returns the Unauthorized (401) response code.

Normal response codes: 204

Error response codes: unauthorized (401)

16.3.3.1. Request

This table shows the header parameters for the show account metadata request:

Name	Type	Description
X-Auth-Token	String (Required)	Authentication token.
X-Newest	Boolean (Optional)	If set to <code>true</code> , Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to <code>true</code> is more expensive for the back end, use it only when it is absolutely needed.

Name	Type	Description
X-Trans-Id-Extra	UUID (Optional)	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the show account metadata request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.

This operation does not accept a request body.

16.3.3.2. Response

This operation does not return a response body.

16.4. Containers

Lists objects in a container. Creates, shows details for, and deletes containers. Creates, updates, shows, and deletes container metadata.

Method	URI	Description
GET	/v1/{account}/{container}{?limit, marker, end_marker, prefix, format, delimiter, path}	Shows details for a container and lists objects, sorted by name, in the container.
PUT	/v1/{account}/{container}	Creates a container.
DELETE	/v1/{account}/{container}	Deletes an empty container.
POST	/v1/{account}/{container}	Creates, updates, or deletes custom metadata for a container.
HEAD	/v1/{account}/{container}	Shows container metadata, including the number of objects and the total bytes of all objects stored in the container.

16.4.1. Show container details and list objects

Method	URI	Description
GET	/v1/{account}/{container}{?limit, marker, end_marker, prefix, format, delimiter, path}	Shows details for a container and lists objects, sorted by name, in the container.

Specify query parameters in the request to filter the list and return a subset of object names. Omit query parameters to return the complete list of object names that are stored in the container, up to 10,000 names. The 10,000 maximum value is configurable. To view the value for the cluster, issue a **GET** /info request.

Example requests and responses:

- OK (200). Success. The response body lists the objects.
- No Content (204). Success. The response body shows no objects. Either the container has no objects or you are paging through a long list of names by using the marker, limit, or end_marker query parameter and you have reached the end of the list.

If the container does not exist, the call returns the Not Found (404) response code.

The operation returns the Range Not Satisfiable (416) response code for any ranged **GET** requests that specify more than:

- Fifty ranges.
- Three overlapping ranges.
- Eight non-increasing ranges.

Normal response codes: 200204

Error response codes: NotFound (404), 416

16.4.1.1. Request

This table shows the header parameters for the show container details and list objects request:

Name	Type	Description
X-Auth-Token	String (Required)	Authentication token.
X-Newest	Boolean (Optional)	If set to true, Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to true is more expensive for the back end, use it only when it is absolutely needed.
Accept	String (Optional)	Instead of using the format query parameter, set this header to application/json, application/xml, or text/xml.
X-Container-Meta-Temp-URL-Key	String (Optional)	The secret key value for temporary URLs.

Name	Type	Description
X-Container-Meta-Temp-URL-Key-2	String (Optional)	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.
X-Trans-Id-Extra	UUID (Optional)	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the show container details and list objects request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (/) character because this character delimits the container and object name. For example, /account/container/object.

This table shows the query parameters for the show container details and list objects request:

Name	Type	Description
limit	Int (Optional)	For an integer value <i>n</i> , limits the number of results to <i>n</i> .
marker	String (Optional)	For a string value, <i>x</i> , returns container names that are greater than the marker value.
end_marker	String (Optional)	For a string value, <i>x</i> , returns container names that are less than the marker value.
prefix	String (Optional)	Prefix value. Named items in the response begin with this value.
format	String (Optional)	The response format. Valid values are <code>json</code>, <code>xml</code>, or <code>plain</code>. The default is <code>plain</code>. If you append the <code>format=xml</code> or <code>format=json</code> query parameter to the storage account URL, the response shows extended container information serialized in that format. If you append the <code>format=plain</code> query parameter, the response lists the container names separated by newlines.

Name	Type	Description
delimiter	Char (Optional)	Delimiter value, which returns the object names that are nested in the container. If you do not set a prefix and set the delimiter to "/" you may get unexpected results where all the objects are returned instead of only those with the delimiter set.
path	String (Optional)	For a string value, returns the object names that are nested in the pseudo path.

This operation does not accept a request body.

16.4.1.2. Response

Example 16.6. Show container details response: HTTP and JSON

```
HTTP/1.1 200 OK
Content-Length: 341
X-Container-Object-Count: 2
Accept-Ranges: bytes
X-Container-Meta-Book: TomSawyer
X-Timestamp: 1389727543.65372
X-Container-Bytes-Used: 26
Content-Type: application/json; charset=utf-8
X-Trans-Id: tx26377fe5fab74869825d1-0052d6bdf
Date: Wed, 15 Jan 2014 16:57:35 GMT
```

```
[
  {
    "hash": "451e372e48e0f6b1114fa0724aa79fa1",
    "last_modified": "2014-01-15T16:41:49.390270",
    "bytes": 14,
    "name": "goodbye",
    "content_type": "application/octet-stream"
  },
  {
    "hash": "ed076287532e86365e841e92bfc50d8c",
    "last_modified": "2014-01-15T16:37:43.427570",
    "bytes": 12,
    "name": "helloworld",
    "content_type": "application/octet-stream"
  }
]
```

This table shows the body parameters for the show container details and list objects response:

Name	Type	Description
name	String (Required)	The name of the container.
hash	String (Required)	The MD5 checksum value of the object content.
bytes	Int (Required)	The total number of bytes that are stored in Object Storage for the account.
content_type	String (Required)	The content type of the object.

Name	Type	Description
last_modified	DateTime (Required)	The date and time when the object was last modified. The date and time stamp format is ISO 8601: CCYY-MM-DDThh:mm:ss±hh:mm For example, 2015-08-27T09:49:58-05:00. The ±hh:mm value, if included, is the time zone as an offset from UTC. In the previous example, the offset value is -05:00.

Example 16.7. Show container details response: HTTP and XML

```

HTTP/1.1 200 OK
Content-Length: 500
X-Container-Object-Count: 2
Accept-Ranges: bytes
X-Container-Meta-Book: TomSawyer
X-Timestamp: 1389727543.65372
X-Container-Bytes-Used: 26
Content-Type: application/xml; charset=utf-8
X-Trans-Id: txc75ea9a6e66f47d79e0c5-0052d6be76
Date: Wed, 15 Jan 2014 16:59:35 GMT

```

```

<?xml version="1.0" encoding="UTF-8"?>
<container name="marktwain">
  <object>
    <name>goodbye</name>
    <hash>451e372e48e0f6b1114fa0724aa79fa1</hash>
    <bytes>14</bytes>
    <content_type>application/octet-stream</content_type>
    <last_modified>2014-01-15T16:41:49.390270</last_modified>
  </object>
  <object>
    <name>helloworld</name>
    <hash>ed076287532e86365e841e92bfc50d8c</hash>
    <bytes>12</bytes>
    <content_type>application/octet-stream</content_type>
    <last_modified>2014-01-15T16:37:43.427570</last_modified>
  </object>
</container>

```

This operation does not return a response body.

16.4.2. Create container

Method	URI	Description
PUT	/v1/{account}/{container}	Creates a container.

You do not need to check whether a container already exists before issuing a **PUT** operation because the operation is idempotent: It creates a container or updates an existing container, as appropriate.

Example requests and responses:

- Create a container with no metadata:

```
curl -i $publicURL/steven -X PUT -H "Content-Length: 0" -H "X-Auth-Token: $token"
```

```
HTTP/1.1 201 Created
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx7f6b7fa09bc2443a94df0-0052d58b56
Date: Tue, 14 Jan 2014 19:09:10 GMT
```

- Create a container with metadata:

```
curl -i $publicURL/marktwain -X PUT -H "X-Auth-Token: $token" -H "X-Container-Meta-Book: TomSawyer"
```

```
HTTP/1.1 201 Created
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx06021f10fc8642b2901e7-0052d58f37
Date: Tue, 14 Jan 2014 19:25:43 GMT
```

Normal response codes: 201204

16.4.2.1. Request

This table shows the header parameters for the create container request:

Name	Type	Description
X-Auth-Token	String (Required)	Authentication token.
X-Container-Read	String (Optional)	Sets a container access control list (ACL) that grants read access. Container ACLs are available on any Object Storage cluster, and are enabled by container rather than by cluster. To set the container read ACL: \$ curl -X {PUT POST} -i -H "X-Auth-Token: TOKEN" -H \ "X-Container-Read: ACL" STORAGE_URL/CONTAINER For example: \$ curl -X PUT -i \ -H "X-Auth-Token: 01010101" \ -H "X-Container-Read: .r:*" \ http://swift.example.com/v1/AUTH_bob/read_container

Name	Type	Description
		In the command, specify the ACL in the X-Container-Read header, as follows: • <code>.r: *#All referrers.</code> • <code>.r: example.com, swift.example.com#Comma-separated list of referrers.</code> • <code>.r listings#Container listing access.</code> • <code>AUTH_username#Access to a user who authenticates through a legacy or non-OpenStack-Identity-based authentication system.</code> • <code>LDAP_#Access to all users who authenticate through an LDAP-based legacy or non-OpenStack-Identity-based authentication system.</code>
X-Container-Write	String (Optional)	Sets an ACL that grants write access.
X-Container-Sync-To	String (Optional)	Sets the destination for container synchronization. Used with the secret key indicated in the X-Container-Sync-Key header. If you want to stop a container from synchronizing, send a blank value for the X-Container-Sync-Key header.
X-Container-Sync-Key	String (Optional)	Sets the secret key for container synchronization. If you remove the secret key, synchronization is halted.
X-Versions-Location	String (Optional)	Enables versioning on this container. The value is the name of another container. You must UTF-8-encode and then URL-encode the name before you include it in the header. To disable versioning, set the header to an empty string.
X-Container-Meta-name	String (Optional)	The container metadata, where {name} is the name of metadata item. You must specify an X-Container-Meta-{name} header for each metadata item (for each {name}) that you want to add or update.
X-Container-Meta-Access-Control-Allow-Origin	String (Optional)	Originating URLs allowed to make cross-origin requests (CORS), separated by spaces. This heading applies to the container only, and all objects within the container with this header applied are CORS-enabled for the allowed origin URLs. A browser (user-agent) typically issues a preflighted request, which is an OPTIONS call that verifies the origin is allowed to make the request. The Object Storage service returns 200 if the originating URL is listed in this header parameter, and issues a 401 if the originating URL is not allowed to make a cross-origin request. Once a 200 is returned, the browser makes a second request to the Object Storage service to retrieve the CORS-enabled object.
X-Container-Meta-Access-Control-Max-Age	String (Optional)	Maximum time for the origin to hold the preflight results. A browser may make an OPTIONS call to verify the origin is allowed to make the request. Set the value to an integer number of seconds after the time that the request was received.
X-Container-Meta-Access-Control-Expose-Headers	String (Optional)	Headers the Object Storage service exposes to the browser (technically, through the user-agent setting), in the request response, separated by spaces. By default the Object Storage service returns the following values for this header: • All “simple response headers” as listed on http://www.w3.org/TR/cors/#simple-response-header. • The headers <code>etag</code>, <code>x-timestamp</code>, <code>x-trans-id</code>.

Name	Type	Description
		All metadata headers (X-Container-Meta-* for containers and X-Object-Meta-* for objects) headers listed in X-Container-Meta-Access-Control-Expose-Headers.
Content-Type	String (Optional)	Changes the MIME type for the object.
X-Detect-Content-Type	Boolean (Optional)	If set to true, Object Storage guesses the content type based on the file extension and ignores the value sent in the Content-Type header, if present.
X-Container-Meta-Temp-URL-Key	String (Optional)	The secret key value for temporary URLs.
X-Container-Meta-Temp-URL-Key-2	String (Optional)	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.
X-Trans-Id-Extra	UUID (Optional)	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the create container request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (/) character because this character delimits the container and object name. For example, /account/container/object.

This operation does not accept a request body.

16.4.2.2. Response

This operation does not return a response body.

16.4.3. Delete container

Method	URI	Description
DELETE	/v1/{account}/{container}	Deletes an empty container.

This operation fails unless the container is empty. An empty container has no objects.

Delete the steven container:

```
curl -i $publicURL/steven -X DELETE -H "X-Auth-Token: $token"
```

If the container does not exist, the response is:

```
HTTP/1.1 404 Not Found
Content-Length: 70
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx4d728126b17b43b598bf7-0052d81e34
Date: Thu, 16 Jan 2014 18:00:20 GMT
```

If the container exists and the deletion succeeds, the response is:

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txf76c375ebece4df19c84c-0052d81f14
Date: Thu, 16 Jan 2014 18:04:04 GMT
```

If the container exists but is not empty, the response is:

```
HTTP/1.1 409 Conflict
Content-Length: 95
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx7782dc6a97b94a46956b5-0052d81f6b
Date: Thu, 16 Jan 2014 18:05:31 GMT

<html><h1>Conflict</h1><p>There was a conflict when trying to complete your
request.</p></html>
```

Normal response codes: 204

Error response codes: NotFound (404), Conflict (409)

16.4.3.1. Request

This table shows the header parameters for the delete container request:

Name	Type	Description
X-Auth-Token	String (Required)	Authentication token.
X-Container-Meta-Temp-URL-Key	String (Optional)	The secret key value for temporary URLs.
X-Container-Meta-Temp-URL-Key-2	String (Optional)	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.

Name	Type	Description
X-Trans-Id-Extra	UUID (Optional)	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the delete container request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (/) character because this character delimits the container and object name. For example, /account/container/object.

This operation does not accept a request body.

16.4.3.2. Response

This operation does not return a response body.

16.4.4. Create, update, or delete container metadata

Method	URI	Description
POST	/v1/{account}/{container}	Creates, updates, or deletes custom metadata for a container.

To create, update, or delete a custom metadata item, use the `X-Container-Meta-{name}` header, where `{name}` is the name of the metadata item.

Subsequent requests for the same key and value pair overwrite the previous value.

To delete container metadata, send an empty value for that header, such as for the `X-Container-Meta-Book` header. If the tool you use to communicate with Object Storage, such as an older version of `cURL`, does not support empty headers, send the `X-Remove-Container-Meta-{name}` header with an arbitrary value. For example, `X-Remove-Container-Meta-Book: x`. The operation ignores the arbitrary value.

If the container already has other custom metadata items, a request to create, update, or delete metadata does not affect those items.

Example requests and responses:

- Create container metadata:

```
curl -i $publicURL/marktwain -X POST -H "X-Auth-Token: $token" -H "X-Container-Meta-Author: MarkTwain" -H "X-Container-Meta-Web-Directory-Type: text/directory" -H "X-Container-Meta-Century: Nineteenth"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx05dbd434c651429193139-0052d82635
Date: Thu, 16 Jan 2014 18:34:29 GMT
```

- Update container metadata:

```
curl -i $publicURL/marktwain -X POST -H "X-Auth-Token: $token" -H "X-Container-Meta-Author: SamuelClemens"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txe60c7314bf614bb39dfe4-0052d82653
Date: Thu, 16 Jan 2014 18:34:59 GMT
```

- Delete container metadata:

```
curl -i $publicURL/marktwain -X POST -H "X-Auth-Token: $token" -H "X-Remove-Container-Meta-Century: x"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx7997e18da2a34a9e84ceb-0052d826d0
```


Date: Thu, 16 Jan 2014 18:37:04 GMT

If the request succeeds, the operation returns the No Content (204) response code.

To confirm your changes, issue a show container metadata request.

Normal response codes: 204

16.4.4.1. Request

This table shows the header parameters for the create, update, or delete container metadata request:

Name	Type	Description
X-Auth-Token	String (Required)	Authentication token.
X-Container-Read	String (Optional)	Sets a container access control list (ACL) that grants read access. Container ACLs are available on any Object Storage cluster, and are enabled by container rather than by cluster. To set the container read ACL: <pre>\$ curl -X {PUT POST} -i -H "X-Auth-Token: TOKEN" -H \ "X-Container-Read: ACL" STORAGE_URL/CONTAINER</pre> For example: <pre>\$ curl -X PUT -i \ -H "X-Auth-Token: 0101010101" \ -H "X-Container-Read: .r:*" \ http://swift.example.com/v1/AUTH_bob/read_container</pre> In the command, specify the ACL in the X-Container-Read header, as follows: .r:*#All referrers. .r:example.com,swift.example.com#Comma-separated list of referrers. .rlistings#Container listing access. - AUTH_username#Access to a user who authenticates through a legacy or non-OpenStack-Identity-based authentication system. - LDAP_#Access to all users who authenticate through an LDAP-based legacy or non-OpenStack-Identity-based authentication system.
X-Remove-Container-name	String (Optional)	Removes the metadata item named {name}. For example, X-Remove-Container-Read removes the X-Container-Read metadata item.
X-Container-Write	String (Optional)	Sets an ACL that grants write access.
X-Container-Sync-To	String (Optional)	Sets the destination for container synchronization. Used with the secret key indicated in the X-Container-Sync-Key header. If you want to stop a container from synchronizing, send a blank value for the X-Container-Sync-Key header.
X-Container-Sync-Key	String (Optional)	Sets the secret key for container synchronization. If you remove the secret key, synchronization is halted.
X-Versions-Location	String (Optional)	Enables versioning on this container. The value is the name of another container. You must UTF-8-encode and then URL-encode the name before you include it in the header. To disable versioning, set the header to an empty string.

Name	Type	Description
X-Remove-Versions-Location	String (Optional)	Set to any value to disable versioning.
X-Container-Meta-name	String (Optional)	The container metadata, where {name} is the name of metadata item. You must specify an X-Container-Meta-{name} header for each metadata item (for each {name}) that you want to add or update.
X-Container-Meta-Access-Control-Allow-Origin	String (Optional)	Originating URLs allowed to make cross-origin requests (CORS), separated by spaces. This heading applies to the container only, and all objects within the container with this header applied are CORS-enabled for the allowed origin URLs. A browser (user-agent) typically issues a preflighted request , which is an OPTIONS call that verifies the origin is allowed to make the request. The Object Storage service returns 200 if the originating URL is listed in this header parameter, and issues a 401 if the originating URL is not allowed to make a cross-origin request. Once a 200 is returned, the browser makes a second request to the Object Storage service to retrieve the CORS-enabled object.
X-Container-Meta-Access-Control-Max-Age	String (Optional)	Maximum time for the origin to hold the preflight results. A browser may make an OPTIONS call to verify the origin is allowed to make the request. Set the value to an integer number of seconds after the time that the request was received.
X-Container-Meta-Access-Control-Expose-Headers	String (Optional)	Headers the Object Storage service exposes to the browser (technically, through the user-agent setting), in the request response, separated by spaces. By default the Object Storage service returns the following values for this header: • All "simple response headers" as listed on http://www.w3.org/TR/cors/#simple-response-header. • The headers etag, x-timestamp, x-trans-id. • All metadata headers (X-Container-Meta-* for containers and X-Object-Meta-* for objects) headers listed in X-Container-Meta-Access-Control-Expose-Headers.
X-Container-Meta-Quota-Bytes	String (Optional)	Sets maximum size of the container, in bytes. Typically these values are set by an administrator. Returns a 413 response (request entity too large) when an object PUT operation exceeds this quota value.
X-Container-Meta-Quota-Count	String (Optional)	Sets maximum object count of the container. Typically these values are set by an administrator. Returns a 413 response (request entity too large) when an object PUT operation exceeds this quota value.
X-Container-Meta-Web-Directory-Type	String (Optional)	Sets the content-type of directory marker objects. If the header is not set, default is application/directory. Directory marker objects are 0-byte objects that represent directories to create a simulated hierarchical structure. For example, if you set "X-Container-Meta-Web-Directory-Type: text/directory", Object Storage treats 0-byte objects with a content-type of text/directory as directories rather than objects.
Content-Type	String (Optional)	Changes the MIME type for the object.
X-Detect-Content-Type	Boolean (Optional)	If set to true, Object Storage guesses the content type based on the file extension and ignores the value sent in the Content-Type header, if present.
X-Container-Meta-Temp-URL-Key	String	The secret key value for temporary URLs.

Name	Type	Description
	(Optional)	
X-Container-Meta-Temp-URL-Key-2	String (Optional)	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.
X-Trans-Id-Extra	UUID (Optional)	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the create, update, or delete container metadata request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (/) character because this character delimits the container and object name. For example, /account/container/object.

This operation does not accept a request body.

16.4.4.2. Response

This operation does not return a response body.

16.4.5. Show container metadata

Method	URI	Description
HEAD	/v1/{account}/{container}	Shows container metadata, including the number of objects and the total bytes of all objects stored in the container.

Show container metadata request:

```
curl -i $publicURL/marktwain -X HEAD -H "X-Auth-Token: $token"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
X-Container-Object-Count: 1
Accept-Ranges: bytes
X-Container-Meta-Book: TomSawyer
X-Timestamp: 1389727543.65372
X-Container-Meta-Author: SamuelClemens
X-Container-Bytes-Used: 14
Content-Type: text/plain; charset=utf-8
X-Trans-Id: tx0287b982a268461b9ec14-0052d826e2
Date: Thu, 16 Jan 2014 18:37:22 GMT
```

If the request succeeds, the operation returns the No Content (204) response code.

Normal response codes: 204

16.4.5.1. Request

This table shows the header parameters for the show container metadata request:

Name	Type	Description
X-Auth-Token	String (Optional)	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
X-Newest	Boolean (Optional)	If set to <code>true</code> , Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to <code>true</code> is more expensive for the back end, use it only when it is absolutely needed.
X-Container-Meta-Temp-URL-Key	String (Optional)	The secret key value for temporary URLs.
X-Container-Meta-Temp-URL-Key-2	String (Optional)	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.
X-Trans-Id-Extra	UUID (Optional)	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header.

Name	Type	Description
		For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the show container metadata request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (/) character because this character delimits the container and object name. For example, /account/container/object.

This operation does not accept a request body.

16.4.5.2. Response

This operation does not return a response body.

16.5. Objects

Creates, replaces, shows details for, and deletes objects. Copies objects from another object with a new or different name. Updates object metadata.

Method	URI	Description
GET	/v1/{account}/{container}/{object} {?temp_url_sig,temp_url_expires, filename,multipart-manifest}	Downloads the object content and gets the object metadata.
PUT	/v1/{account}/{container}/{object} {?multipart-manifest,temp_url_sig, temp_url_expires,filename}	Creates an object with data content and metadata, or replaces an existing object with data content and metadata.
COPY	/v1/{account}/{container}/{object}	Copies an object to another object in the object store.
DELETE	/v1/{account}/{container}/{object} {?multipart-manifest}	Permanently deletes an object from the object store.
HEAD	/v1/{account}/{container}/{object} {?temp_url_sig,temp_url_expires, filename}	Shows object metadata.
POST	/v1/{account}/{container}/{object}	Creates or updates object metadata.

16.5.1. Get object content and metadata

Method	URI	Description
GET	/v1/{account}/{container}/{object}{?temp_url_sig,temp_url_expires,filename,multipart-manifest}	Downloads the object content and gets the object metadata.

This operation returns the object metadata in the response headers and the object content in the response body.

If this is a large object, the response body contains the concatenated content of the segment objects. To get the manifest instead of concatenated segment objects for a static large object, use the `multipart-manifest` query parameter.

Example requests and responses:

- Show object details for the goodbye object in the marktwain container:

```
curl -i $publicURL/marktwain/goodbye -X GET -H "X-Auth-Token: $token"
```

```
HTTP/1.1 200 OK
Content-Length: 14
Accept-Ranges: bytes
Last-Modified: Wed, 15 Jan 2014 16:41:49 GMT
Etag: 451e372e48e0f6b1114fa0724aa79fa1
X-Timestamp: 1389804109.39027
X-Object-Meta-Orig-Filename: goodbyeworld.txt
Content-Type: application/octet-stream
X-Trans-Id: tx8145a190241f4cf6b05f5-0052d82a34
Date: Thu, 16 Jan 2014 18:51:32 GMT

Goodbye World!
```

- Show object details for the goodbye object, which does not exist, in the janeausten container:

```
curl -i $publicURL/janeausten/goodbye -X GET -H "X-Auth-Token: $token"
```

```
HTTP/1.1 404 Not Found
Content-Length: 70
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx073f7cbb850c4c99934b9-0052d82b04
Date: Thu, 16 Jan 2014 18:55:00 GMT

<html><h1>Not Found</h1><p>The resource could not be found.</p></html>
```

The operation returns the `Range Not Satisfiable` (416) response code for any ranged **GET** requests that specify more than:

- Fifty ranges.
- Three overlapping ranges.
- Eight non-increasing ranges.

Normal response codes: 200

Error response codes: NotFound (404), 416

16.5.1.1. Request

This table shows the header parameters for the get object content and metadata request:

Name	Type	Description
X-Auth-Token	String (Optional)	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
X-Newest	Boolean (Optional)	If set to <code>true</code> , Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to <code>true</code> is more expensive for the back end, use it only when it is absolutely needed.
Range	String (Optional)	The ranges of content to get. You can use the Range header to get portions of data by using one or more range specifications. To specify many ranges, separate the range specifications with a comma. The types of range specifications are: • Byte range specification. Use <code>FIRST_BYTE_OFFSET</code> to specify the start of the data range, and <code>LAST_BYTE_OFFSET</code> to specify the end. You can omit the <code>LAST_BYTE_OFFSET</code> and if you do, the value defaults to the offset of the last byte of data. • Suffix byte range specification. Use <code>LENGTH</code> bytes to specify the length of the data range. The following forms of the header specify the following ranges of data: • Range: <code>bytes=-5</code>. The last five bytes. • Range: <code>bytes=10-15</code>. The five bytes of data after a 10-byte offset. • Range: <code>bytes=10-15, -5</code>. A multi-part response that contains the last five bytes and the five bytes of data after a 10-byte offset. The Content-Type response header contains <code>multipart/byteranges</code>. • Range: <code>bytes=4-6</code>. Bytes 4 to 6 inclusive. • Range: <code>bytes=2-2</code>. Byte 2, the third byte of the data. • Range: <code>bytes=6-</code>. Byte 6 and after. • Range: <code>bytes=1-3, 2-5</code>. A multi-part response that contains bytes 1 to 3 inclusive, and bytes 2 to 5 inclusive. The Content-Type response header contains <code>multipart/byteranges</code>.
If-Match	String (Optional)	See Request for Comments: 2616 .
If-None-Match	String (Optional)	In combination with <code>Expect: 100-Continue</code> , specify an "If-None-Match: *" header to query whether the server already has a copy of the object before any data is sent.
If-Modified-Since	Date (Optional)	See Request for Comments: 2616 .
If-Unmodified-Since	Date	See Request for Comments: 2616 .

Name	Type	Description
	<i>(Optional)</i>	
X-Trans-Id-Extra	UUID <i>(Optional)</i>	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the get object content and metadata request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (/) character because this character delimits the container and object name. For example, /account/container/object.
{object}	String	The unique name for the object.

This table shows the query parameters for the get object content and metadata request:

Name	Type	Description
temp_url_sig	String <i>(Required)</i>	Used with temporary URLs to sign the request with an HMAC-SHA1 cryptographic signature that defines the allowed HTTP method, expiration date, full path to the object, and the secret key for the temporary URL. For more information about temporary URLs, see Temporary URL middleware .
temp_url_expires	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the signature for temporary URLs expires. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT. For more information about temporary URLs, see Temporary URL middleware.
filename	String <i>(Optional)</i>	Overrides the default file name. Object Storage generates a default file name for GET temporary URLs that is based on the object name. Object Storage returns this value in the Content-Disposition response header. Browsers can interpret this file name value as a file attachment to save. For more information about temporary URLs, see Temporary URL middleware .
multipart-manifest	String <i>(Optional)</i>	If you include the multipart-manifest=get query parameter and the object is a large object, the object contents are not returned. Instead, the manifest is returned in the X-Object-Manifest response

Name	Type	Description
		header for dynamic large objects or in the response body for static large objects.

This operation does not accept a request body.

16.5.1.2. Response

This operation does not return a response body.

16.5.2. Create or replace object

Method	URI	Description
PUT	/v1/{account}/{container}/{object} {?multipart-manifest,temp_url_sig, temp_url_expires,filename}	Creates an object with data content and metadata, or replaces an existing object with data content and metadata.

The **PUT** operation always creates an object. If you use this operation on an existing object, you replace the existing object and metadata rather than modifying the object. Consequently, this operation returns the Created (201) response code.

If you use this operation to copy a manifest object, the new object is a normal object and not a copy of the manifest. Instead it is a concatenation of all the segment objects. This means that you cannot copy objects larger than 5 GB.

Example requests and responses:

- Create object:

```
curl -i $publicURL/janeastten/helloworld.txt -X PUT -H "Content-Length: 1" -H "Content-Type: text/html; charset=UTF-8" -H "X-Auth-Token: $token"
```

```
HTTP/1.1 201 Created
Last-Modified: Fri, 17 Jan 2014 17:28:35 GMT
Content-Length: 116
Etag: d41d8cd98f00b204e9800998ecf8427e
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx4d5e4f06d357462bb732f-0052d96843
Date: Fri, 17 Jan 2014 17:28:35 GMT
```

- Replace object:

```
curl -i $publicURL/janeastten/helloworld -X PUT -H "Content-Length: 0" -H "X-Auth-Token: $token"
```

```
HTTP/1.1 201 Created
Last-Modified: Fri, 17 Jan 2014 17:28:35 GMT
Content-Length: 116
Etag: d41d8cd98f00b204e9800998ecf8427e
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx4d5e4f06d357462bb732f-0052d96843
Date: Fri, 17 Jan 2014 17:28:35 GMT
```

The Created (201) response code indicates a successful write.

If the request times out, the operation returns the Request Timeout (408) response code.

The Length Required (411) response code indicates a missing Transfer-Encoding or Content-Length request header.

If the MD5 checksum of the data that is written to the object store does not match the optional ETag value, the operation returns the Unprocessable Entity (422) response code.

Normal response codes: 201

Error response codes: timeout (408), lengthRequired (411), unprocessableEntity (422)

16.5.2.1. Request

This table shows the header parameters for the create or replace object request:

Name	Type	Description
X-Object-Manifest	String (Optional)	Set to specify that this is a dynamic large object manifest object. The value is the container and object name prefix of the segment objects in the form <code>container/prefix</code> . You must UTF-8-encode and then URL-encode the names of the container and prefix before you include them in this header.
X-Auth-Token	String (Optional)	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
Content-Length	Int (Optional)	Set to the length of the object content. Do not set if chunked transfer encoding is being used.
Transfer-Encoding	String (Optional)	Set to chunked to enable chunked transfer encoding. If used, do not set the Content-Length header to a non-zero value.
Content-Type	String (Optional)	Changes the MIME type for the object.
X-Detect-Content-Type	Boolean (Optional)	If set to true, Object Storage guesses the content type based on the file extension and ignores the value sent in the Content-Type header, if present.
X-Copy-From	String (Optional)	If set, this is the name of an object used to create the new object by copying the X-Copy-From object. The value is in form <code>{container}/{object}</code> . You must UTF-8-encode and then URL-encode the names of the container and object before you include them in the header. Using PUT with X-Copy-From has the same effect as using the COPY operation to copy an object. Using Range header with X-Copy-From will create a new partial copied object with bytes set by Range.
ETag	String (Optional)	The MD5 checksum value of the request body. For example, the MD5 checksum value of the object content. You are strongly recommended to compute the MD5 checksum value of object content and include it in the request. This enables the Object Storage API to check the integrity of the upload. The value is not quoted.
Content-Disposition	String (Optional)	If set, specifies the override behavior for the browser. For example, this header might specify that the browser use a download program to save this file rather than show the file, which is the default.
Content-Encoding	String (Optional)	If set, the value of the Content-Encoding metadata.
X-Delete-At	Int (Optional)	The date and time in UNIX Epoch time stamp format when the system removes the object. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Delete-After	Int (Optional)	The number of seconds after which the system removes the object. Internally, the Object Storage system stores this value in the X-Delete-At metadata item.
X-Object-Meta-name	String	The object metadata, where {name} is the name of the metadata item.

Name	Type	Description
	<i>(Optional)</i>	You must specify an X-Object-Meta- {name} header for each meta-data {name} item that you want to add or update.
If-None-Match	String <i>(Optional)</i>	In combination with Expect: 100-Continue, specify an "If-None-Match: *" header to query whether the server already has a copy of the object before any data is sent.
X-Trans-Id-Extra	UUID <i>(Optional)</i>	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the create or replace object request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (/) character because this character delimits the container and object name. For example, /account/container/object.
{object}	String	The unique name for the object.

This table shows the query parameters for the create or replace object request:

Name	Type	Description
multipart-manifest	String <i>(Optional)</i>	If ?multipart-manifest=put, the object is a static large object manifest and the body contains the manifest.
temp_url_sig	String <i>(Required)</i>	Used with temporary URLs to sign the request with an HMAC-SHA1 cryptographic signature that defines the allowed HTTP method, expiration date, full path to the object, and the secret key for the temporary URL. For more information about temporary URLs, see Temporary URL middleware .
temp_url_expires	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the signature for temporary URLs expires. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT. For more information about temporary URLs, see Temporary URL middleware.
filename	String	Overrides the default file name. Object Storage generates a default file name for GET temporary URLs that is based on the object name.

Name	Type	Description
	<i>(Optional)</i>	Object Storage returns this value in the Content-Disposition response header. Browsers can interpret this file name value as a file attachment to save. For more information about temporary URLs, see Temporary URL middleware .

This operation does not accept a request body.

16.5.2.2. Response

This operation does not return a response body.

16.5.3. Copy object

Method	URI	Description
COPY	/v1/{account}/{container}/{object}	Copies an object to another object in the object store.

You can copy an object to a new object with the same name. Copying to the same name is an alternative to using **POST** to add metadata to an object. With **POST**, you must specify all the metadata. With **COPY**, you can add additional metadata to the object.

With **COPY**, you can set the X-Fresh-Metadata header to `true` to copy the object without any existing metadata.

Alternatively, you can use **PUT** with the X-Copy-From request header to accomplish the same operation as the **COPY** object operation.

The **PUT** operation always creates an object. If you use this operation on an existing object, you replace the existing object and metadata rather than modifying the object. Consequently, this operation returns the Created (201) response code.

If you use this operation to copy a manifest object, the new object is a normal object and not a copy of the manifest. Instead it is a concatenation of all the segment objects. This means that you cannot copy objects larger than 5 GB in size. All metadata is preserved during the object copy. If you specify metadata on the request to copy the object, either **PUT** or **COPY**, the metadata overwrites any conflicting keys on the target (new) object.

Example requests and responses:

- Copy the goodbye object from the marktwain container to the janeausten container:

```
curl -i $publicURL/marktwain/goodbye -X COPY -H "X-Auth-Token: $token" -H
"Destination: janeausten/goodbye"
```

```
HTTP/1.1 201 Created
Content-Length: 0
X-Copied-From-Last-Modified: Thu, 16 Jan 2014 21:19:45 GMT
X-Copied-From: marktwain/goodbye
Last-Modified: Fri, 17 Jan 2014 18:22:57 GMT
Etag: 451e372e48e0f6b1114fa0724aa79fa1
Content-Type: text/html; charset=UTF-8
X-Object-Meta-Movie: AmericanPie
X-Trans-Id: txdc481ad49d24e9a81107-0052d97501
Date: Fri, 17 Jan 2014 18:22:57 GMT
```

- Alternatively, you can use **PUT** to copy the goodbye object from the marktwain container to the janeausten container. This request requires a Content-Length header, even if it is set to zero (0).

```
curl -i $publicURL/janeausten/goodbye -X PUT -H "X-Auth-Token: $token" -H
"X-Copy-From: /marktwain/goodbye" -H "Content-Length: 0"
```

```
HTTP/1.1 201 Created
```

```

Content-Length: 0
X-Copied-From-Last-Modified: Thu, 16 Jan 2014 21:19:45 GMT
X-Copied-From: marktwain/goodbye
Last-Modified: Fri, 17 Jan 2014 18:22:57 GMT
Etag: 451e372e48e0f6b1114fa0724aa79fa1
Content-Type: text/html; charset=UTF-8
X-Object-Meta-Movie: AmericanPie
X-Trans-Id: txdc481ad49d24e9a81107-0052d97501
Date: Fri, 17 Jan 2014 18:22:57 GMT

```

When several replicas exist, the system copies from the most recent replica. That is, the **COPY** operation behaves as though the X-Newest header is in the request.

Normal response codes: 201

16.5.3.1. Request

This table shows the header parameters for the copy object request:

Name	Type	Description
X-Auth-Token	String (Optional)	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
Destination	String (Required)	The container and object name of the destination object in the form of /container/object. You must UTF-8-encode and then URL-encode the names of the destination container and object before you include them in this header.
Content-Type	String (Optional)	Changes the MIME type for the object.
Content-Encoding	String (Optional)	If set, the value of the Content-Encoding metadata.
Content-Disposition	String (Optional)	If set, specifies the override behavior for the browser. For example, this header might specify that the browser use a download program to save this file rather than show the file, which is the default.
X-Object-Meta-name	String (Optional)	The object metadata, where {name} is the name of the metadata item. You must specify an X-Object-Meta-{name} header for each metadata {name} item that you want to add or update.
X-Fresh-Metadata	Boolean (Optional)	Enables object creation that omits existing user metadata. If set to true, the COPY request creates an object without existing user metadata. Default value is false.
X-Trans-Id-Extra	UUID (Optional)	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra

Name	Type	Description
		request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the copy object request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (/) character because this character delimits the container and object name. For example, /account/container/object.
{object}	String	The unique name for the object.

This operation does not accept a request body.

16.5.3.2. Response

This operation does not return a response body.

16.5.4. Delete object

Method	URI	Description
DELETE	/v1/{account}/{container}/{object} {?multipart-manifest}	Permanently deletes an object from the object store.

You can use the **COPY** method to copy the object to a new location. Then, use the **DELETE** method to delete the original object.

Object deletion occurs immediately at request time. Any subsequent **GET**, **HEAD**, **POST**, or **DELETE** operations return a 404 Not Found error code.

For static large object manifests, you can add the `?multipart-manifest=delete` query parameter. This operation deletes the segment objects and if all deletions succeed, this operation deletes the manifest object.

Example request and response:

- Delete the `helloworld` object from the `marktwain` container:

```
curl -i $publicURL/marktwain/helloworld -X DELETE -H "X-Auth-Token: $token"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx36c7606fcd1843f59167c-0052d6fdac
Date: Wed, 15 Jan 2014 21:29:16 GMT
```

Typically, the **DELETE** operation does not return a response body. However, with the `multipart-manifest=delete` query parameter, the response body contains a list of manifest and segment objects and the status of their **DELETE** operations.

Normal response codes: 204

16.5.4.1. Request

This table shows the header parameters for the delete object request:

Name	Type	Description
X-Auth-Token	String (Optional)	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
X-Trans-Id-Extra	UUID (Optional)	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment

Name	Type	Description
		and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the delete object request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (/) character because this character delimits the container and object name. For example, /account/container/object.
{object}	String	The unique name for the object.

This table shows the query parameters for the delete object request:

Name	Type	Description
multipart-manifest	String <i>(Optional)</i>	If you include the multipart-manifest=delete query parameter and the object is a static large object, the segment objects and manifest object are deleted. If you omit the multipart-manifest=delete query parameter and the object is a static large object, the manifest object is deleted but the segment objects are not deleted. For a bulk delete, the response body looks the same as it does for a normal bulk delete. In contrast, a plain object DELETE response has an empty body.

This operation does not accept a request body.

16.5.4.2. Response

This operation does not return a response body.

16.5.5. Show object metadata

Method	URI	Description
HEAD	/v1/{account}/{container}/{object} {?temp_url_sig,temp_url_expires, filename}	Shows object metadata.

If the Content-Length response header is non-zero, the example cURL command stalls after it prints the response headers because it is waiting for a response body. However, the Object Storage system does not return a response body for the **HEAD** operation.

Example requests and responses:

- Show object metadata:

```
curl -i $publicURL/marktwain/goodbye -X HEAD -H "X-Auth-Token: $token"
```

```
HTTP/1.1 200 OK
Content-Length: 14
Accept-Ranges: bytes
Last-Modified: Thu, 16 Jan 2014 21:12:31 GMT
Etag: 451e372e48e0f6b1114fa0724aa79fa1
X-Timestamp: 1389906751.73463
X-Object-Meta-Book: GoodbyeColumbus
Content-Type: application/octet-stream
X-Trans-Id: tx37ea34dcd1ed48ca9bc7d-0052d84b6f
Date: Thu, 16 Jan 2014 21:13:19 GMT
```

If the request succeeds, the operation returns the 200 response code.

Normal response codes: 200204

16.5.5.1. Request

This table shows the header parameters for the show object metadata request:

Name	Type	Description
X-Auth-Token	String (Required)	Authentication token.
X-Newest	Boolean (Optional)	If set to true, Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to true is more expensive for the back end, use it only when it is absolutely needed.
X-Trans-Id-Extra	UUID (Optional)	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header.

Name	Type	Description
		For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the show object metadata request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (/) character because this character delimits the container and object name. For example, /account/container/object.
{object}	String	The unique name for the object.

This table shows the query parameters for the show object metadata request:

Name	Type	Description
temp_url_sig	String (Required)	Used with temporary URLs to sign the request with an HMAC-SHA1 cryptographic signature that defines the allowed HTTP method, expiration date, full path to the object, and the secret key for the temporary URL. For more information about temporary URLs, see Temporary URL middleware .
temp_url_expires	Int (Required)	The date and time in UNIX Epoch time stamp format when the signature for temporary URLs expires. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT. For more information about temporary URLs, see Temporary URL middleware.
filename	String (Optional)	Overrides the default file name. Object Storage generates a default file name for GET temporary URLs that is based on the object name. Object Storage returns this value in the Content-Disposition response header. Browsers can interpret this file name value as a file attachment to save. For more information about temporary URLs, see Temporary URL middleware .

This operation does not accept a request body.

16.5.5.2. Response

This operation does not return a response body.

16.5.6. Create or update object metadata

Method	URI	Description
POST	/v1/{account}/{container}/{object}	Creates or updates object metadata.

To create or update custom metadata, use the `X-Object-Meta-{name}` header, where `{name}` is the name of the metadata item.

In addition to the custom metadata, you can update the `Content-Type`, `Content-Encoding`, `Content-Disposition`, and `X-Delete-At` system metadata items. However you cannot update other system metadata, such as `Content-Length` or `Last-Modified`.

You can use **COPY** as an alternate to the **POST** operation by copying to the same object. With the **POST** operation you must specify all metadata items, whereas with the **COPY** operation, you need to specify only changed or additional items.

All metadata is preserved during the object copy. If you specify metadata on the request to copy the object, either **PUT** or **COPY**, the metadata overwrites any conflicting keys on the target (new) object.

A **POST** request deletes any existing custom metadata that you added with a previous **PUT** or **POST** request. Consequently, you must specify all custom metadata in the request. However, system metadata is unchanged by the **POST** request unless you explicitly supply it in a request header.

You can also set the `X-Delete-At` or `X-Delete-After` header to define when to expire the object.

When used as described in this section, the **POST** operation creates or replaces metadata. This form of the operation has no request body.

You can also use the [form POST feature](#) to upload objects.

Example requests and responses:

- Create object metadata:

```
curl -i $publicURL/marktwain/goodbye -X POST -H "X-Auth-Token: $token" -H
  "X-Object-Meta-Book: GoodbyeColumbus"
```

```
HTTP/1.1 202 Accepted
Content-Length: 76
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txb5fb5c91ba1f4f37bb648-0052d84b3f
Date: Thu, 16 Jan 2014 21:12:31 GMT

<html><h1>Accepted</h1><p>The request is accepted for processing.</p></html>
```

- Update object metadata:

```
curl -i $publicURL/marktwain/goodbye -X POST -H "X-Auth-Token: $token" -H "X-Object-Meta-Book: GoodbyeOldFriend"
```

```

HTTP/1.1 202 Accepted
Content-Length: 76
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx5ec7ab81cdb34ced887c8-0052d84ca4
Date: Thu, 16 Jan 2014 21:18:28 GMT

<html><h1>Accepted</h1><p>The request is accepted for processing.</p></html>

```

Normal response codes: 202

16.5.6.1. Request

This table shows the header parameters for the create or update object metadata request:

Name	Type	Description
X-Auth-Token	String (Optional)	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
X-Object-Meta-name	String (Optional)	The object metadata, where {name} is the name of the metadata item. You must specify an X-Object-Meta-{name} header for each metadata {name} item that you want to add or update.
X-Delete-At	Int (Optional)	The date and time in UNIX Epoch time stamp format when the system removes the object. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
Content-Disposition	String (Optional)	If set, specifies the override behavior for the browser. For example, this header might specify that the browser use a download program to save this file rather than show the file, which is the default.
Content-Encoding	String (Optional)	If set, the value of the Content-Encoding metadata.
X-Delete-After	Int (Optional)	The number of seconds after which the system removes the object. Internally, the Object Storage system stores this value in the X-Delete-At metadata item.
Content-Type	String (Optional)	Changes the MIME type for the object.
X-Detect-Content-Type	Boolean (Optional)	If set to true, Object Storage guesses the content type based on the file extension and ignores the value sent in the Content-Type header, if present.
X-Trans-Id-Extra	UUID (Optional)	Extra transaction information. Use the X-Trans-Id-Extra request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the X-Trans-Id-Extra request header value to the transaction ID value in the generated X-Trans-Id response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the X-Trans-Id-Extra request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the X-Trans-Id-Extra request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs.

Name	Type	Description
		You can also use X-Trans-Id-Extra strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the create or update object metadata request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (/) character because this character delimits the container and object name. For example, /account/container/object.
{object}	String	The unique name for the object.

This operation does not accept a request body.

16.5.6.2. Response

This operation does not return a response body.

17. Orchestration API v1 (CURRENT)

Uses a template language to orchestrate OpenStack services.

Method	URI	Description
General API information		
API versions		
GET	/	Lists all Orchestration API versions.
Stacks		
POST	/v1/{tenant_id}/stacks	Creates a stack.
POST	/v1/{tenant_id}/stacks	Creates a stack from existing resources.
GET	/v1/{tenant_id}/stacks{?id, status, name, action, tenant, username, owner_id, limit, marker, show_deleted, show_nested, sort_keys, tags, tags_any, not_tags, not_tags_any, sort_dir, global_tenant, with_count}	Lists active stacks.
POST	/v1/{tenant_id}/stacks/preview	Previews a stack.
PUT	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/preview	Previews an update for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}	Finds the canonical URL for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/resources	Finds the canonical URL for a resource list of a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Shows details for a stack.
PUT	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Updates a stack.
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Deletes a stack and its snapshots.
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/abandon	Deletes a stack but leaves its resources intact, and returns data that describes the stack and its resources.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots	Takes a snapshot of all resources in a stack. All snapshots are deleted when the stack is deleted.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots	Lists snapshots for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}	Shows details for a snapshot.
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}	Deletes a stack snapshot.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}/restore	Restores a stack snapshot.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs	Lists outputs for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs/{output_key}	Shows details for a stack output.
Stack actions		

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Suspends a stack.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Resumes a suspended stack.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Cancels a currently running update of a stack.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Checks whether the resources are in expected states for a stack.
Stack resources		
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources{?nested_depth,with_detail}	Lists resources in a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}	Shows data for a resource.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/metadata	Shows metadata for a resource.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/signal	Sends a signal to a resource.
Stack events		
GET	/v1/{tenant_id}/stacks/{stack_name}/events	Finds the canonical URL for the event list of a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/events{?resource_action,resource_status,resource_name,resource_type,limit,marker,sort_keys,sort_dir}	Lists events for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events{?resource_action,resource_status,resource_type,limit,marker,sort_keys,sort_dir}	Lists events for a stack resource.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events/{event_id}	Shows details for an event.
Templates		
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/template	Gets a template for a stack.
GET	/v1/{tenant_id}/template_versions	Lists all available template versions.
POST	/v1/{tenant_id}/validate	Validates a template.
GET	/v1/{tenant_id}/resource_types/{type_name}/template{?template_type}	Shows the template representation for a resource type.
GET	/v1/{tenant_id}/resource_types/{type_name}	Shows the interface schema for a resource type.
GET	/v1/{tenant_id}/resource_types{?name,version,support_status}	Lists all supported template resource types.
Build info		
GET	/v1/{tenant_id}/build_info	Shows build information for an Orchestration deployment.
Software configuration		

Method	URI	Description
POST	/v1/{tenant_id}/software_configs	Creates a software configuration.
GET	/v1/{tenant_id}/software_configs/{config_id}	Shows details for a software configuration.
DELETE	/v1/{tenant_id}/software_configs/{config_id}	Deletes a software configuration.
GET	/v1/{tenant_id}/software_deployments	Lists all available software deployments.
POST	/v1/{tenant_id}/software_deployments	Creates a software deployment.
GET	/v1/{tenant_id}/software_deployments/metadata/{server_id}	Shows the deployment configuration metadata for a server.
GET	/v1/{tenant_id}/software_deployments/{deployment_id}	Shows details for a software deployment.
PUT	/v1/{tenant_id}/software_deployments/{deployment_id}	Updates a software deployment.
DELETE	/v1/{tenant_id}/software_deployments/{deployment_id}	Deletes a software deployment.
Manage service		
GET	/v1/{tenant_id}/services	Enables administrative users to view details for all orchestration engines.

17.1. General API information

Authenticated calls that target a known URI but that use an HTTP method that the implementation does not support return a 405 `Method Not Allowed` error code. In addition, the HTTP `OPTIONS` method is supported for each known URI. In both cases, the `Allow` response header indicates the HTTP methods that are supported for the resource.

17.2. API versions

Method	URI	Description
GET	/	Lists all Orchestration API versions.

17.2.1. List versions

Method	URI	Description
GET	/	Lists all Orchestration API versions.

Normal response codes: 200

17.2.1.1. Request

This operation does not accept a request body.

17.2.1.2. Response

Example 17.1. List versions: JSON response

```
{
  "versions": [
    {
      "status": "CURRENT",
      "id": "v1.0",
      "links": [
        {
          "href": "http://23.253.228.211:8000/v1/",
          "rel": "self"
        }
      ]
    }
  ]
}
```

17.3. Stacks

Method	URI	Description
POST	/v1/{tenant_id}/stacks	Creates a stack.
POST	/v1/{tenant_id}/stacks	Creates a stack from existing resources.
GET	/v1/{tenant_id}/stacks{?id, status, name, action, tenant, username, owner_id, limit, marker, show_deleted, show_nested, sort_keys, tags, tags_any, not_tags, not_tags_any, sort_dir, global_tenant, with_count}	Lists active stacks.
POST	/v1/{tenant_id}/stacks/preview	Previews a stack.
PUT	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/preview	Previews an update for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}	Finds the canonical URL for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/resources	Finds the canonical URL for a resource list of a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Shows details for a stack.
PUT	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Updates a stack.

Method	URI	Description
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Deletes a stack and its snapshots.
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/abandon	Deletes a stack but leaves its resources intact, and returns data that describes the stack and its resources.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots	Takes a snapshot of all resources in a stack. All snapshots are deleted when the stack is deleted.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots	Lists snapshots for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}	Shows details for a snapshot.
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}	Deletes a stack snapshot.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}/restore	Restores a stack snapshot.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs	Lists outputs for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs/{output_key}	Shows details for a stack output.

17.3.1. Create stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks	Creates a stack.

This table shows the possible response codes for this operation:

Response Code	Name	Description
201		
400	Bad Request	
401	Unauthorized	
409	Conflict	
500	Internal Server Error	

17.3.1.1. Request

This table shows the URI parameters for the create stack request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 17.2. Create stack: JSON request

```
{
  "files": {},
  "disable_rollback": true,
  "parameters": {
    "flavor": "m1.heat"
  },
  "stack_name": "teststack",
  "template": {
    "heat_template_version": "2013-05-23",
    "description": "Simple template to test heat commands",
    "parameters": {
      "flavor": {
        "default": "m1.tiny",
        "type": "string"
      }
    }
  },
  "resources": {
    "hello_world": {
      "type": "OS::Nova::Server",
      "properties": {
        "key_name": "heat_key",
        "flavor": {
          "get_param": "flavor"
        },
        "image": "40be8d1a-3eb9-40de-8abd-43237517384f",
        "user_data": "#!/bin/bash -xv\\necho \\\"hello world\\\" &gt; /
root/hello-world.txt\\n"
      }
    }
  }
}
```

```
    }  
  },  
  "timeout_mins": 60  
}
```

17.3.1.2. Response

Example 17.3. Create stack: JSON response

```
{  
  "stack": {  
    "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",  
    "links": [  
      {  
        "href": "http://192.168.123.200:8004/v1/  
eb1c63a4f77141548385f113a28f0f52/stacks/teststack/3095aefc-09fb-4bc7-b1f0-  
f21a304e864c",  
        "rel": "self"  
      }  
    ]  
  }  
}
```

17.3.2. Adopt stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks	Creates a stack from existing resources.

This table shows the possible response codes for this operation:

Response Code	Name	Description
201		
400	Bad Request	
401	Unauthorized	
409	Conflict	
500	Internal Server Error	

17.3.2.1. Request

This table shows the URI parameters for the adopt stack request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 17.4. Adopt stack: JSON request

```
{
  "adopt_stack_data": {
    "action": "CREATE",
    "id": "bxxxxx4-0xx2-4xx1-axx6-xxxxxxxc",
    "name": "teststack",
    "resources": {
      "MyServer": {
        "action": "CREATE",
        "metadata": {},
        "name": "MyServer",
        "resource_data": {},
        "resource_id": "cxxxx3-dxx3-4xx-bxx2-3xxxxxxxxxa",
        "status": "COMPLETE",
        "type": "OS::Trove::Instance"
      }
    },
    "status": "COMPLETE",
    "template": {
      "heat_template_version": "2013-05-23",
      "resources": {
        "MyServer": {
          "type": "OS::Trove::Instance",
          "properties": {
            "flavor": "m1.small",
            "size": 10
          }
        }
      }
    }
  }
}
```

```
    },
    "stack_name": "{stack_name}",
    "timeout_mins": "{timeout_mins}"
}
```

17.3.2.2. Response

Example 17.5. Adopt stack: JSON response

```
{
  "stack": {
    "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
    "links": [
      {
        "href": "http://192.168.123.200:8004/v1/
eb1c63a4f77141548385f113a28f0f52/stacks/teststack/3095aefc-09fb-4bc7-b1f0-
f21a304e864c",
        "rel": "self"
      }
    ]
  }
}
```


17.3.3. List stack data

Method	URI	Description
GET	/v1/{tenant_id}/stacks{?id, status, name, action, tenant, username, owner_id, limit, marker, show_deleted, show_nested, sort_keys, tags, tags_any, not_tags, not_tags_any, sort_dir, global_tenant, with_count}	Lists active stacks.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
500	Internal Server Error	

17.3.3.1. Request

This table shows the URI parameters for the list stack data request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This table shows the query parameters for the list stack data request:

Name	Type	Description
id	UUID (Optional)	Filters the stack list by a stack ID. Use this filter multiple times to filter by multiple IDs.
status	String (Optional)	Filters the stack list by a status. Use this filter multiple times to filter by multiple statuses.
name	String (Optional)	Filters the stack list by a name. Use this filter multiple times to filter by multiple names.
action	String (Optional)	Filters the stack list by an action. Use this filter multiple times to filter by multiple actions.
tenant	String (Optional)	Filters the stack list by a tenant. Use this filter multiple times to filter by multiple tenants.
username	String (Optional)	Filters the stack list by a user name. Use this filter multiple times to filter by multiple user names.
owner_id	UUID (Optional)	Filters the stack list by an owner ID, which is the ID of the parent stack of listed stack. Use this filter multiple times to filter by multiple owner IDs.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and

Name	Type	Description
		use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
show_deleted	Boolean (Optional)	Set to <code>true</code> to include deleted stacks in the list. Default is <code>false</code> , which excludes deleted stacks from the list.
show_nested	Boolean (Optional)	Set to <code>true</code> to include nested stacks in the list. Default is <code>false</code> , which excludes nested stacks from the list.
sort_keys	String (Optional)	Sorts the stack list by <code>stack_name</code> , <code>stack_status</code> , <code>creation_time</code> , or <code>updated_time</code> key.
tags	String (Optional)	Lists stacks that contain one or more simple string tags. To specify multiple tags, separate the tags with commas. For example, <code>tag1, tag2</code> . The boolean AND expression is used to combine multiple tags.
tags_any	String (Optional)	Lists stacks that contain one or more simple string tags. To specify multiple tags, separate the tags with commas. For example, <code>tag1, tag2</code> . The boolean OR expression is used to combine multiple tags.
not_tags	String (Optional)	Lists stacks that do not contain one or more simple string tags. To specify multiple tags, separate the tags with commas. For example, <code>tag1, tag2</code> . The boolean AND expression is used to combine multiple tags.
not_tags_any	String (Optional)	Lists stacks that do not contain one or more simple string tags. To specify multiple tags, separate the tags with commas. For example, <code>tag1, tag2</code> . The boolean OR expression is used to combine multiple tags.
sort_dir	String (Optional)	The sort direction of the list. A valid value is <code>asc</code> (ascending) or <code>desc</code> (descending).
global_tenant	Boolean (Optional)	Set to <code>true</code> to include stacks from all tenants in the stack list. Specify policy requirements in the <code>Orchestration policy.json</code> file. Default is <code>false</code> .
with_count	Boolean (Optional)	Set to <code>true</code> to include a count key in the response. The count key value is the number of stacks that match the query criteria. Default is <code>false</code> .

This operation does not accept a request body.

17.3.3.2. Response

Example 17.6. List stack data: JSON response

```
{
  "stacks": [
    {
      "creation_time": "2014-06-03T20:59:46Z",
      "description": "sample stack",
      "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
      "links": [
        {
          "href": "http://192.168.123.200:8004/v1/eb1c63a4f77141548385f113a28f0f52/stacks/simple_stack/3095aefc-09fb-4bc7-b1f0-f21a304e864c",
          "rel": "self"
        }
      ]
    }
  ]
}
```

```
    ],  
    "stack_name": "simple_stack",  
    "stack_status": "CREATE_COMPLETE",  
    "stack_status_reason": "Stack CREATE completed successfully",  
    "updated_time": "",  
    "tags": ""  
  }  
]  
}
```

17.3.4. Preview stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks/preview	Previews a stack.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
409	Conflict	
500	Internal Server Error	

17.3.4.1. Request

This table shows the URI parameters for the preview stack request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 17.7. Preview stack: JSON request

```
{
  "files": {},
  "disable_rollback": true,
  "parameters": {
    "flavor": "m1.heat"
  },
  "stack_name": "teststack",
  "template": {
    "heat_template_version": "2013-05-23",
    "description": "Simple template to test heat commands",
    "parameters": {
      "flavor": {
        "default": "m1.tiny",
        "type": "string"
      }
    },
    "resources": {
      "hello_world": {
        "type": "OS::Nova::Server",
        "properties": {
          "key_name": "heat_key",
          "flavor": {
            "get_param": "flavor"
          },
          "image": "40be8d1a-3eb9-40de-8abd-43237517384f",
          "user_data": "#!/bin/bash -xv\\necho \\\"hello world\\\" &gt; /
root/hello-world.txt\\n"
        }
      }
    }
  },
}
```

```

    "timeout_mins": 60
}

```

17.3.4.2. Response

Example 17.8. Preview stack: JSON response

```

{
  "stack": {
    "capabilities": [],
    "creation_time": "2015-01-31T15:12:36Z",
    "description": "HOT template for Nova Server resource.\n",
    "disable_rollback": true,
    "id": "None",
    "links": [
      {
        "href": "http://192.168.122.102:8004/v1/6e18cc2bdb48a5basad2dc499f6804/stacks/test_stack/None",
        "rel": "self"
      }
    ],
    "notification_topics": [],
    "parameters": {
      "OS::project_id": "6e18cc2bdb48a5basad2dc499f6804",
      "OS::stack_id": "None",
      "OS::stack_name": "teststack",
      "admin_user": "cloud-user",
      "flavor": "m1.small",
      "image": "F20-cfg",
      "key_name": "heat_key",
      "server_name": "MyServer"
    },
    "parent": null,
    "resources": [
      {
        "attributes": {},
        "description": "",
        "metadata": {},
        "physical_resource_id": "",
        "properties": {
          "description": "Ping and SSH",
          "name": "the_sg",
          "rules": [
            {
              "direction": "ingress",
              "ethertype": "IPv4",
              "port_range_max": null,
              "port_range_min": null,
              "protocol": "icmp",
              "remote_group_id": null,
              "remote_ip_prefix": null,
              "remote_mode": "remote_ip_prefix"
            },
            {
              "direction": "ingress",
              "ethertype": "IPv4",
              "port_range_max": 65535,
              "port_range_min": 1,
              "protocol": "tcp",
              "remote_group_id": null,

```

```

        "remote_ip_prefix": null,
        "remote_mode": "remote_ip_prefix"
    },
    {
        "direction": "ingress",
        "ethertype": "IPv4",
        "port_range_max": 65535,
        "port_range_min": 1,
        "protocol": "udp",
        "remote_group_id": null,
        "remote_ip_prefix": null,
        "remote_mode": "remote_ip_prefix"
    }
]
},
"required_by": [
    "server1"
],
"resource_action": "INIT",
"resource_identity": {
    "path": "/resources/the_sg_res",
    "stack_id": "None",
    "stack_name": "teststack",
    "tenant": "6e18cc2bdb48a5b3cad2dc499f6804"
},
"resource_name": "the_sg_res",
"resource_status": "COMPLETE",
"resource_status_reason": "",
"resource_type": "OS::Neutron::SecurityGroup",
"stack_identity": {
    "path": "",
    "stack_id": "None",
    "stack_name": "teststack",
    "tenant": "6e18cc2bdb48a5b3cad2dc499f6804"
},
"stack_name": "teststack",
"updated_time": "2015-01-31T15:12:36Z"
},
{
    "attributes": {
        "accessIPv4": "",
        "accessIPv6": "",
        "addresses": "",
        "console_urls": "",
        "first_address": "",
        "instance_name": "",
        "name": "MyServer",
        "networks": "",
        "show": ""
    },
    "description": "",
    "metadata": {},
    "physical_resource_id": "",
    "properties": {
        "admin_pass": null,
        "admin_user": "cloud-user",
        "availability_zone": null,
        "block_device_mapping": null,
        "config_drive": null,
        "diskConfig": null,

```

```

        "flavor": "m1.small",
        "flavor_update_policy": "RESIZE",
        "image": "F20-cfg",
        "image_update_policy": "REPLACE",
        "key_name": "heat_key",
        "metadata": {
            "ha_stack": "None"
        },
        "name": "MyServer",
        "networks": [
            {
                "fixed_ip": null,
                "network": "private",
                "port": null,
                "uuid": null
            }
        ],
        "personality": {},
        "reservation_id": null,
        "scheduler_hints": null,
        "security_groups": [
            "None"
        ],
        "software_config_transport": "POLL_SERVER_CFN",
        "user_data": "",
        "user_data_format": "HEAT_CFNTOOLS"
    },
    "required_by": [],
    "resource_action": "INIT",
    "resource_identity": {
        "path": "/resources/hello_world",
        "stack_id": "None",
        "stack_name": "teststack",
        "tenant": "6e18cc2bdb48a3433cad2dc499sdf32234"
    },
    "resource_name": "hello_world",
    "resource_status": "COMPLETE",
    "resource_status_reason": "",
    "resource_type": "OS::Nova::Server",
    "stack_identity": {
        "path": "",
        "stack_id": "None",
        "stack_name": "teststack",
        "tenant": "6e18cc2bdb48a3433cad2dc499sdf32234"
    },
    "stack_name": "teststack",
    "updated_time": "2015-01-31T15:12:36Z"
}
},
"stack_name": "test_stack",
"stack_owner": "admin",
"template_description": "HOT template for Nova Server resource.\n",
"timeout_mins": null,
"updated_time": null
}
}

```

17.3.5. Preview stack update

Method	URI	Description
PUT	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/preview	Previews an update for a stack.

Normal response codes: 200

17.3.5.1. Request

This table shows the URI parameters for the preview stack update request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

Example 17.9. Preview stack update: JSON request

```
{
  "template": {
    "heat_template_version": "2013-05-23",
    "description": "Create a simple stack",
    "parameters": {
      "flavor": {
        "default": "m1.tiny",
        "type": "string"
      }
    },
    "resources": {
      "hello_world": {
        "type": "OS::Nova::Server",
        "properties": {
          "key_name": "heat_key",
          "flavor": {
            "get_param": "flavor"
          },
          "image": "40be8d1a-3eb9-40de-8abd-43237517384f",
          "user_data": "#!/bin/bash -xv\\necho \\\"hello world\\\" > /
root/hello-world.txt\\n"
        }
      }
    },
    "parameters": {
      "flavor": "m1.small"
    }
  }
}
```

17.3.5.2. Response

Example 17.10. Preview stack update: JSON response

```
{
  "unchanged": [
    {
```



```

        "updated_time": "datetime",
        "resource_name": "",
        "physical_resource_id": "{resource id or ''}",
        "resource_action": "CREATE",
        "resource_status": "COMPLETE",
        "resource_status_reason": "",
        "resource_type": "restype",
        "stack_identity": "{stack_id}",
        "stack_name": "{stack_name}"
    }
],
"updated": [
    {
        "updated_time": "datetime",
        "resource_name": "",
        "physical_resource_id": "{resource id or ''}",
        "resource_action": "CREATE",
        "resource_status": "COMPLETE",
        "resource_status_reason": "",
        "resource_type": "restype",
        "stack_identity": "{stack_id}",
        "stack_name": "{stack_name}"
    }
],
"replaced": [
    {
        "updated_time": "datetime",
        "resource_name": "",
        "physical_resource_id": "{resource id or ''}",
        "resource_action": "CREATE",
        "resource_status": "COMPLETE",
        "resource_status_reason": "",
        "resource_type": "restype",
        "stack_identity": "{stack_id}",
        "stack_name": "{stack_name}"
    }
],
"added": [
    {
        "updated_time": "datetime",
        "resource_name": "",
        "physical_resource_id": "{resource id or ''}",
        "resource_action": "CREATE",
        "resource_status": "COMPLETE",
        "resource_status_reason": "",
        "resource_type": "restype",
        "stack_identity": "{stack_id}",
        "stack_name": "{stack_name}"
    }
],
"deleted": [
    {
        "updated_time": "datetime",
        "resource_name": "",
        "physical_resource_id": "{resource id or ''}",
        "resource_action": "CREATE",
        "resource_status": "COMPLETE",
        "resource_status_reason": "",
        "resource_type": "restype",
        "stack_identity": "{stack_id}",

```

```
        "stack_name": "{stack_name}"
    }
]
```

17.3.6. Find stack

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}	Finds the canonical URL for a stack.

Also works with verbs other than **GET**, so that you can perform **PUT** and **DELETE** operations on a current stack. Set your client to follow redirects. When redirecting, the request method should not change as defined in RFC2626. However, in many clients the default behavior is to change the method to **GET** when you receive a 302 response code because this behavior is ubiquitous in web browsers.

This table shows the possible response codes for this operation:

Response Code	Name	Description
302		
400	Bad Request	
401	Unauthorized	
404	Not Found	
500	Internal Server Error	

17.3.6.1. Request

This table shows the URI parameters for the find stack request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.

This operation does not accept a request body.

17.3.7. Find stack resources

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/resources	Finds the canonical URL for a resource list of a stack.

The canonical URL is returned for only non-deleted stacks. To fetch the resource list for deleted stacks, use the following endpoint:

```
/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources
```

Normal response codes: 302

17.3.7.1. Request

This table shows the URI parameters for the find stack resources request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.

This operation does not accept a request body.

17.3.8. Show stack details

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Shows details for a stack.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	
500	Internal Server Error	

17.3.8.1. Request

This table shows the URI parameters for the show stack details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

This operation does not accept a request body.

17.3.8.2. Response

Example 17.11. Show stack details: JSON response

```
{
  "stack": {
    "capabilities": [],
    "creation_time": "2014-06-03T20:59:46Z",
    "description": "sample stack",
    "disable_rollback": true,
    "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
    "links": [
      {
        "href": "http://192.168.123.200:8004/v1/eb1c63a4f77141548385f113a28f0f52/stacks/simple_stack/3095aefc-09fb-4bc7-b1f0-f21a304e864c",
        "rel": "self"
      }
    ],
    "notification_topics": [],
    "outputs": [],
    "parameters": {
      "OS::project_id": "3ab5b02f-a01f-4f95-afa1-e254afc4a435",
      "OS::stack_id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
      "OS::stack_name": "simple_stack"
    }
  }
}
```

```
    },  
    "stack_name": "simple_stack",  
    "stack_owner": "simple_username",  
    "stack_status": "CREATE_COMPLETE",  
    "stack_status_reason": "Stack CREATE completed successfully",  
    "template_description": "sample stack",  
    "stack_user_project_id": "65728b74-cfe7-4f17-9c15-11d4f686e591",  
    "timeout_mins": "",  
    "updated_time": "",  
    "parent": "",  
    "tags": ""  
  }  
}
```

17.3.9. Update stack

Method	URI	Description
PUT	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Updates a stack.

This table shows the possible response codes for this operation:

Response Code	Name	Description
202		
400	Bad Request	
401	Unauthorized	
404	Not Found	
500	Internal Server Error	

17.3.9.1. Request

This table shows the URI parameters for the update stack request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

Example 17.12. Update stack: JSON request

```
{
  "template": {
    "heat_template_version": "2013-05-23",
    "description": "Create a simple stack",
    "parameters": {
      "flavor": {
        "default": "m1.tiny",
        "type": "string"
      }
    },
    "resources": {
      "hello_world": {
        "type": "OS::Nova::Server",
        "properties": {
          "key_name": "heat_key",
          "flavor": {
            "get_param": "flavor"
          },
          "image": "40be8d1a-3eb9-40de-8abd-43237517384f",
          "user_data": "#!/bin/bash -xv\\necho \\\"hello world\\\" > /
root/hello-world.txt\\n"
        }
      }
    }
  },
  "parameters": {
```

```
        "flavor": "m1.small"  
    }  
}
```


17.3.10. Delete stack

Method	URI	Description
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Deletes a stack and its snapshots.

This table shows the possible response codes for this operation:

Response Code	Name	Description
204		
400	Bad Request	
401	Unauthorized	
404	Not Found	
500	Internal Server Error	

17.3.10.1. Request

This table shows the URI parameters for the delete stack request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

This operation does not accept a request body.

17.3.11. Abandon stack

Method	URI	Description
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/abandon	Deletes a stack but leaves its resources intact, and returns data that describes the stack and its resources.

This method can be disabled from the server side. If it is disabled, this call throws an exception.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	
500	Internal Server Error	

17.3.11.1. Request

This table shows the URI parameters for the abandon stack request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

This operation does not accept a request body.

17.3.11.2. Response

Example 17.13. Abandon stack: JSON response

```
{
  "status": "COMPLETE",
  "name": "g",
  "dry_run": true,
  "template": {
    "outputs": {
      "instance_ip": {
        "value": {
          "str_replace": {
            "params": {
              "username": "ec2-user",
              "hostname": {
                "get_attr": [
                  "server",
                  "first_address"
                ]
              }
            }
          }
        }
      }
    }
  },
}
```

```

        "template": "ssh username@hostname"
    }
}
},
"heat_template_version": "2013-05-23",
"resources": {
    "server": {
        "type": "OS::Nova::Server",
        "properties": {
            "key_name": {
                "get_param": "key_name"
            },
            "image": {
                "get_param": "image"
            },
            "flavor": {
                "get_param": "flavor"
            }
        }
    }
},
"parameters": {
    "key_name": {
        "default": "heat_key",
        "type": "string"
    },
    "image": {
        "default": "fedora-amd64",
        "type": "string"
    },
    "flavor": {
        "default": "m1.small",
        "type": "string"
    }
}
},
"action": "CREATE",
"id": "16934ca3-40e0-4fb2-a289-a700662ec05a",
"resources": {
    "server": {
        "status": "COMPLETE",
        "name": "server",
        "resource_data": {},
        "resource_id": "39d5dad7-7d7a-4cc8-bd84-851e9e2ff4ea",
        "action": "CREATE",
        "type": "OS::Nova::Server",
        "metadata": {}
    }
}
}

```

17.3.12. Snapshot stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots	Takes a snapshot of all resources in a stack. All snapshots are deleted when the stack is deleted.

Normal response codes: 200

17.3.12.1. Request

This table shows the URI parameters for the snapshot stack request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

Example 17.14. Snapshot stack: JSON request

```
{
  "name": "vol_snapshot"
}
```

17.3.12.2. Response

Example 17.15. Snapshot stack: JSON response

```
{
  "id": "13c3a4b5-0585-440e-85a4-6f96b20e7a78",
  "name": "vol_snapshot",
  "status": "IN_PROGRESS",
  "status_reason": null,
  "data": null,
  "creation_time": "2015-09-01T20:57:55Z"
}
```

17.3.13. List snapshots

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots	Lists snapshots for a stack.

Normal response codes: 200

17.3.13.1. Request

This table shows the URI parameters for the list snapshots request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

This operation does not accept a request body.

17.3.13.2. Response

Example 17.16. List snapshots: JSON response

```
{
  "snapshots": [
    {
      "id": "7c4e1ef4-bf1b-41ab-a0c8-ce01f4ffdfa1",
      "name": "vol_snapshot",
      "status": "IN_PROGRESS",
      "status_reason": null,
      "creation_time": "2015-08-04T20:57:55Z",
      "data": null
    }
  ]
}
```

17.3.14. Show snapshot

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}	Shows details for a snapshot.

Normal response codes: 200

17.3.14.1. Request

This table shows the URI parameters for the show snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

17.3.14.2. Response

Example 17.17. Show snapshot: JSON response

```
{
  "snapshot": {
    "id": "7c4e1ef4-bf1b-41ab-a0c8-ce01f4ffdfa1",
    "name": "vol_snapshot",
    "status": "COMPLETE",
    "status_reason": "Stack SNAPSHOT completed successfully",
    "creation_time": "2015-08-04T20:57:55Z",
    "data": {
      "status": "COMPLETE",
      "name": "stack_vol1",
      "stack_user_project_id": "fffa11067b1c48129ddfb78fba2bf09f",
      "environment": {
        "parameters": {},
        "resource_registry": {
          "resources": {}
        }
      },
      "template": {
        "heat_template_version": "2013-05-23",
        "resources": {
          "volume": {
            "type": "OS::Cinder::Volume",
            "properties": {
              "size": 1
            }
          }
        }
      }
    }
  },
}
```

```
"action": "SNAPSHOT",
"project_id": "ecdb08032cd042179692a1b148f6565e",
"id": "656452c2-e151-40da-8704-c844e69b485c",
"resources": {
  "volume": {
    "status": "COMPLETE",
    "name": "volume",
    "resource_data": {
      "backup_id": "99108cf8-398f-461b-a043-bdceb7c9f572"
    },
    "resource_id": "3ab8cf79-807b-4c40-b743-0655f91e072f",
    "action": "SNAPSHOT",
    "type": "OS::Cinder::Volume",
    "metadata": {}
  }
}
}
```

17.3.15. Delete snapshot

Method	URI	Description
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}	Deletes a stack snapshot.

Normal response codes: 204

17.3.15.1. Request

This table shows the URI parameters for the delete snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

17.3.16. Restore snapshot

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}/restore	Restores a stack snapshot.

You can restore only active stacks from a snapshot. You must recreate deleted stacks.

Normal response codes: 202

17.3.16.1. Request

This table shows the URI parameters for the restore snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

17.3.17. List outputs

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs	Lists outputs for a stack.

Normal response codes: 200

17.3.17.1. Request

This table shows the URI parameters for the list outputs request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

This operation does not accept a request body.

17.3.17.2. Response

Example 17.18. List outputs: JSON response

```
{
  "outputs": [
    {
      "output_key": "output name",
      "output_value": "output value",
      "description": "output description",
      "output_error": null
    }
  ]
}
```

17.3.18. Show output

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs/{output_key}	Shows details for a stack output.

Normal response codes: 200

17.3.18.1. Request

This table shows the URI parameters for the show output request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.
{output_key}	String	The key of a stack output.

This operation does not accept a request body.

17.3.18.2. Response

Example 17.19. Show output: JSON response

```
{
  "output": {
    "output_key": "output_name",
    "output_value": "output_value",
    "description": "output description",
    "output_error": null
  }
}
```

17.4. Stack actions

Performs non-lifecycle operations on the stack. Specify the action in the request body.

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Suspends a stack.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Resumes a suspended stack.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Cancel a currently running update of a stack.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Checks whether the resources are in expected states for a stack.

17.4.1. Suspend stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Suspends a stack.

Normal response codes: 200

17.4.1.1. Request

This table shows the URI parameters for the suspend stack request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

Example 17.20. Suspend stack: JSON request

```
{
  "suspend": null
}
```

17.4.2. Resume stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Resumes a suspended stack.

Normal response codes: 200

17.4.2.1. Request

This table shows the URI parameters for the resume stack request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

Example 17.21. Resume stack: JSON request

```
{
  "resume": null
}
```

17.4.3. Cancel stack update

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Cancels a currently running update of a stack.

Normal response codes: 200

17.4.3.1. Request

This table shows the URI parameters for the cancel stack update request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

Example 17.22. Cancel stack update: JSON request

```
{
  "cancel_update": null
}
```

17.4.4. Check stack resources

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Checks whether the resources are in expected states for a stack.

Normal response codes: 200

17.4.4.1. Request

This table shows the URI parameters for the check stack resources request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

Example 17.23. Check stack resources: JSON request

```
{
  "check": null
}
```

17.5. Stack resources

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources{?nested_depth,with_detail}	Lists resources in a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}	Shows data for a resource.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/metadata	Shows metadata for a resource.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/signal	Sends a signal to a resource.

17.5.1. List resources

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources{?nested_depth,with_detail}	Lists resources in a stack.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	

17.5.1.1. Request

This table shows the URI parameters for the list resources request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

This table shows the query parameters for the list resources request:

Name	Type	Description
nested_depth	Int (Optional)	Includes resources from nested stacks up to the nested_depth levels of recursion.
with_detail	Boolean (Optional)	Enables detailed resource information for each resource in list of resources.

This operation does not accept a request body.

17.5.1.2. Response

Example 17.24. List resources: JSON response

Lists resources in a stack, in JSON format.

```
{
  "resources": [
    {
      "creation_time": "2015-06-25T14:59:53",
      "links": [
        {
          "href": "http://hostname/v1/1234/stacks/mystack/629a32d0-ac4f-4f63-b58d-f0d047b1ba4c/resources/random_key_name",
          "rel": "self"
        }
      ]
    }
  ]
}
```



```
        },
        {
            "href": "http://hostname/v1/1234/stacks/mystack/629a32d0-
ac4f-4f63-b58d-f0d047b1ba4c",
            "rel": "stack"
        }
    ],
    "logical_resource_id": "random_key_name",
    "physical_resource_id": "mystack-random_key_name-pmjmy5pks735",
    "required_by": [],
    "resource_name": "random_key_name",
    "resource_status": "CREATE_COMPLETE",
    "resource_status_reason": "state changed",
    "resource_type": "OS::Heat::RandomString",
    "updated_time": "2015-06-25T14:59:53"
}
]
```

17.5.2. Show resource data

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}	Shows data for a resource.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	

17.5.2.1. Request

This table shows the URI parameters for the show resource data request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.
{resource_name}	String	The name of a resource in the stack.

This operation does not accept a request body.

17.5.2.2. Response

Example 17.25. Show resource data: JSON response

Shows resource data for a resource, in JSON format.

```
{
  "resource": {
    "attributes": {
      "value": "I9S20uIp"
    },
    "creation_time": "2015-06-25T14:59:53",
    "description": "",
    "links": [
      {
        "href": "http://hostname/v1/1234/stacks/mystack/629a32d0-ac4f-4f63-b58d-f0d047b1ba4c/resources/random_key_name",
        "rel": "self"
      },
      {
        "href": "http://hostname/v1/1234/stacks/mystack/629a32d0-ac4f-4f63-b58d-f0d047b1ba4c",
        "rel": "stack"
      }
    ]
  }
}
```

```
    ],  
    "logical_resource_id": "random_key_name",  
    "physical_resource_id": "mystack-random_key_name-pmjmy5pks735",  
    "required_by": [],  
    "resource_name": "random_key_name",  
    "resource_status": "CREATE_COMPLETE",  
    "resource_status_reason": "state changed",  
    "resource_type": "OS::Heat::RandomString",  
    "updated_time": "2015-06-25T14:59:53"  
  }  
}
```

17.5.3. Show resource metadata

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/metadata	Shows metadata for a resource.

Normal response codes: 200

17.5.3.1. Request

This table shows the URI parameters for the show resource metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.
{resource_name}	String	The name of a resource in the stack.

This operation does not accept a request body.

17.5.3.2. Response

Example 17.26. Show resource metadata: JSON response

```
{
  "metadata": {
    "some_key": "some_value",
    "some_other_key": "some_other_value"
  }
}
```

17.5.4. Send a signal to a resource

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/signal	Sends a signal to a resource.

The contents of the request body depends on the resource to which you send a signal.

Some resources cannot receive signals. If you send them a signal, they return a 400 error code.

Normal response codes: 200

17.5.4.1. Request

This table shows the URI parameters for the send a signal to a resource request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.
{resource_name}	String	The name of a resource in the stack.

This operation does not accept a request body.

17.6. Stack events

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/events	Finds the canonical URL for the event list of a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/events{?resource_action,resource_status,resource_name,resource_type,limit,marker,sort_keys,sort_dir}	Lists events for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events{?resource_action,resource_status,resource_type,limit,marker,sort_keys,sort_dir}	Lists events for a stack resource.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events/{event_id}	Shows details for an event.

17.6.1. Find stack events

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/events	Finds the canonical URL for the event list of a stack.

Normal response codes: 302

17.6.1.1. Request

This table shows the URI parameters for the find stack events request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.

This operation does not accept a request body.

17.6.2. List stack events

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/events{?resource_action,resource_status,resource_name,resource_type,limit,marker,sort_keys,sort_dir}	Lists events for a stack.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	
500	Internal Server Error	

17.6.2.1. Request

This table shows the URI parameters for the list stack events request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

This table shows the query parameters for the list stack events request:

Name	Type	Description
resource_action	String (Optional)	Filters the event list by a resource action. You can use this filter multiple times to filter by multiple resource actions. Valid resource actions are ADOPT, CHECK, CREATE, DELETE, INIT, RESTORE, RESUME, ROLLBACK, SNAPSHOT, SUSPEND, and UPDATE.
resource_status	String (Optional)	Filters the event list by a resource status. You can use this filter multiple times to filter by multiple resource statuses. Valid resource statuses are COMPLETE, FAILED, and IN_PROGRESS.
resource_name	String (Optional)	Filters the event list by a resource name. You can use this filter multiple times to filter by multiple resource names.
resource_type	String (Optional)	Filters the event list by a resource type. You can use this filter multiple times to filter by multiple resource types. Valid resource types include OS::Cinder::Volume, OS::Nova::Server, OS::Neutron::Port, and so on.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the

Name	Type	Description
		response as the marker parameter value in a subsequent limited request.
sort_keys	String (Optional)	Sorts the list by the resource_type or created_at key.
sort_dir	String (Optional)	The sort direction of the list. A valid value is asc (ascending) or desc (descending).

This operation does not accept a request body.

17.6.2.2. Response

Example 17.27. List stack events: JSON response

```
{
  "events": [
    {
      "resource_name": "port",
      "event_time": "2014-07-23T08:14:47Z",
      "links": [
        {
          "href": "http://192.168.123.200:8004/v1/dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-aeb1-396ff1d151c5/resources/port/events/474bdfd0-a450-46ec-a78a-0c7faa404073",
          "rel": "self"
        },
        {
          "href": "http://192.168.123.200:8004/v1/dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-aeb1-396ff1d151c5/resources/port",
          "rel": "resource"
        },
        {
          "href": "http://192.168.123.200:8004/v1/dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-aeb1-396ff1d151c5",
          "rel": "stack"
        }
      ],
      "logical_resource_id": "port",
      "resource_status": "CREATE_FAILED",
      "resource_status_reason": "NotFound: Subnet f8a699d0-3537-429e-87a5-6b5a8d0c2bf0 could not be found",
      "physical_resource_id": null,
      "id": "474bdfd0-a450-46ec-a78a-0c7faa404073"
    },
    {
      "resource_name": "port",
      "event_time": "2014-07-23T08:14:47Z",
      "links": [
        {
          "href": "http://192.168.123.200:8004/v1/dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-aeb1-396ff1d151c5/resources/port/events/66fa95b6-e6f8-4f05-b1af-e828f5aba04c",
          "rel": "self"
        },
        {

```



```
        "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5/resources/port",
        "rel": "resource"
    },
    {
        "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5",
        "rel": "stack"
    }
],
"logical_resource_id": "port",
"resource_status": "CREATE_IN_PROGRESS",
"resource_status_reason": "state changed",
"physical_resource_id": null,
"id": "66fa95b6-e6f8-4f05-b1af-e828f5aba04c"
}
]
```

17.6.3. List resource events

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events{?resource_action,resource_status,resource_type,limit,marker,sort_keys,sort_dir}	Lists events for a stack resource.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	

17.6.3.1. Request

This table shows the URI parameters for the list resource events request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.
{resource_name}	String	The name of a resource in the stack.

This table shows the query parameters for the list resource events request:

Name	Type	Description
resource_action	String (Optional)	Filters the event list by a resource action. You can use this filter multiple times to filter by multiple resource actions. Valid resource actions are ADOPT, CHECK, CREATE, DELETE, INIT, RESTORE, RESUME, ROLLBACK, SNAPSHOT, SUSPEND, and UPDATE.
resource_status	String (Optional)	Filters the event list by a resource status. You can use this filter multiple times to filter by multiple resource statuses. Valid resource statuses are COMPLETE, FAILED, and IN_PROGRESS.
resource_type	String (Optional)	Filters the event list by a resource type. You can use this filter multiple times to filter by multiple resource types. Valid resource types include OS::Cinder::Volume, OS::Nova::Server, OS::Neutron::Port, and so on.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
marker	String (Optional)	The ID of the last-seen item. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.
sort_keys	String	Sorts the list by the resource_type or created_at key.

Name	Type	Description
	<i>(Optional)</i>	
sort_dir	String <i>(Optional)</i>	The sort direction of the list. A valid value is asc (ascending) or desc (descending).

This operation does not accept a request body.

17.6.3.2. Response

Example 17.28. List resource events: JSON response

```
{
  "events": [
    {
      "resource_name": "port",
      "event_time": "2014-07-23T08:14:47Z",
      "links": [
        {
          "href": "http://192.168.123.200:8004/v1/dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-aeb1-396ff1d151c5/resources/port/events/474bdfd0-a450-46ec-a78a-0c7faa404073",
          "rel": "self"
        },
        {
          "href": "http://192.168.123.200:8004/v1/dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-aeb1-396ff1d151c5/resources/port",
          "rel": "resource"
        },
        {
          "href": "http://192.168.123.200:8004/v1/dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-aeb1-396ff1d151c5",
          "rel": "stack"
        }
      ],
      "logical_resource_id": "port",
      "resource_status": "CREATE_FAILED",
      "resource_status_reason": "NotFound: Subnet f8a699d0-3537-429e-87a5-6b5a8d0c2bf0 could not be found",
      "physical_resource_id": null,
      "id": "474bdfd0-a450-46ec-a78a-0c7faa404073"
    },
    {
      "resource_name": "port",
      "event_time": "2014-07-23T08:14:47Z",
      "links": [
        {
          "href": "http://192.168.123.200:8004/v1/dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-aeb1-396ff1d151c5/resources/port/events/66fa95b6-e6f8-4f05-b1af-e828f5aba04c",
          "rel": "self"
        },
        {
          "href": "http://192.168.123.200:8004/v1/dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-aeb1-396ff1d151c5/resources/port",
          "rel": "resource"
        }
      ]
    }
  ]
}
```

```
    },
    {
      "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5",
      "rel": "stack"
    }
  ],
  "logical_resource_id": "port",
  "resource_status": "CREATE_IN_PROGRESS",
  "resource_status_reason": "state changed",
  "physical_resource_id": null,
  "id": "66fa95b6-e6f8-4f05-b1af-e828f5aba04c"
}
]
```

17.6.4. Show event details

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events/{event_id}	Shows details for an event.

Normal response codes: 200

17.6.4.1. Request

This table shows the URI parameters for the show event details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.
{resource_name}	String	The name of a resource in the stack.
{event_id}	UUID	The UUID of the event that is related to the resource in the stack.

This operation does not accept a request body.

17.6.4.2. Response

Example 17.29. Show event details: JSON response

```
{
  "event": {
    "event_time": "2015-06-25T14:59:53",
    "id": "8db23e2e-72b2-47a2-9ed9-b52417f56e50",
    "links": [
      {
        "href": "http://hostname/v1/1234/stacks/mystack/56789/
resources/random_key_name/events/8db23e2e-72b2-47a2-9ed9-b52417f56e50",
        "rel": "self"
      },
      {
        "href": "http://hostname/v1/1234/stacks/mystack/56789/
resources/random_key_name",
        "rel": "resource"
      },
      {
        "href": "http://hostname/v1/1234/stacks/mystack/56789",
        "rel": "stack"
      }
    ],
    "logical_resource_id": "random_key_name",
    "physical_resource_id": null,
    "resource_name": "random_key_name",
    "resource_properties": {
      "character_classes": null,
      "character_sequences": null,

```

```
        "length": 8,  
        "salt": null,  
        "sequence": null  
    },  
    "resource_status": "CREATE_IN_PROGRESS",  
    "resource_status_reason": "state changed",  
    "resource_type": "OS::Heat::RandomString"  
}
```

17.7. Templates

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/template	Gets a template for a stack.
GET	/v1/{tenant_id}/template_versions	Lists all available template versions.
POST	/v1/{tenant_id}/validate	Validates a template.
GET	/v1/{tenant_id}/resource_types/{type_name}/template{?template_type}	Shows the template representation for a resource type.
GET	/v1/{tenant_id}/resource_types/{type_name}	Shows the interface schema for a resource type.
GET	/v1/{tenant_id}/resource_types{?name,version,support_status}	Lists all supported template resource types.

17.7.1. Get stack template

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/template	Gets a template for a stack.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	
500	Internal Server Error	

17.7.1.1. Request

This table shows the URI parameters for the get stack template request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	UUID	The UUID of the stack.

This operation does not accept a request body.

17.7.1.2. Response

Example 17.30. Get stack template: JSON response

Returns the template for a stack.

```
{
  "description": "Hello world HOT template that just defines a single
server. Contains just base features to verify base HOT support.\n",
  "heat_template_version": "2013-05-23",
  "outputs": {
    "foo": {
      "description": "Show foo parameter value",
      "value": {
        "get_param": "foo"
      }
    }
  },
  "parameters": {
    "foo": {
      "default": "secret",
      "description": "Name of an existing key pair to use for the
server",
      "hidden": true,
      "type": "string"
    }
  }
}
```

```
    }  
  },  
  "resources": {  
    "random_key_name": {  
      "properties": {  
        "length": 8  
      },  
      "type": "OS::Heat::RandomString"  
    }  
  }  
}
```


17.7.2. List template versions

Method	URI	Description
GET	/v1/{tenant_id}/template_versions	Lists all available template versions.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	
500	Internal Server Error	

17.7.2.1. Request

This table shows the URI parameters for the list template versions request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

17.7.2.2. Response

Example 17.31. List template versions: JSON response

```
{
  "template_versions": [
    {
      "version": "heat_template_version.2014-10-16",
      "type": "hot"
    },
    {
      "version": "heat_template_version.2015-04-30",
      "type": "hot"
    },
    {
      "version": "HeatTemplateFormatVersion.2012-12-12",
      "type": "cfn"
    },
    {
      "version": "heat_template_version.2015-10-15",
      "type": "hot"
    },
    {
      "version": "AWSTemplateFormatVersion.2010-09-09",
      "type": "cfn"
    },
    {
      "version": "heat_template_version.2013-05-23",
```

```
        "type": "hot"
      },
      {
        "version": "heat_template_version.2016-04-08",
        "type": "hot"
      }
    ]
  }
```

17.7.3. Validate template

Method	URI	Description
POST	/v1/{tenant_id}/validate	Validates a template.

Normal response codes: 200

17.7.3.1. Request

This table shows the URI parameters for the validate template request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 17.32. Validate template: JSON request

```
{
  "template_url": "/PATH_TO_HEAT_TEMPLATES/WordPress_Single_Instance.
template"
}
```

17.7.3.2. Response

Example 17.33. Validate template: JSON response

```
{
  "Description": "A template that provides a single server instance.",
  "Parameters": {
    "server-size": {
      "default": "1GB Standard Instance",
      "description": "Server size",
      "type": "String",
      "constraints": [
        {
          "allowed_values": [
            "512MB Standard Instance",
            "1GB Standard Instance",
            "4GB Standard Instance",
            "8GB Standard Instance"
          ],
          "description": "Must be a valid server size."
        }
      ]
    },
    "key_name": {
      "description": "Keypair name for SSH access to the server",
      "required": true,
      "type": "String"
    },
    "server_name": {
      "default": "My server",
      "description": "My server",
      "type": "String"
    }
  }
}
```

```
    },  
    "ParameterGroups": [  
      {  
        "label": "Parameter groups",  
        "description": "My parameter groups",  
        "parameters": [  
          "param_name-1",  
          "param_name-2"  
        ]  
      }  
    ]  
  }  
}
```

17.7.4. Show resource template

Method	URI	Description
GET	/v1/{tenant_id}/resource_types/{type_name}/template{?template_type}	Shows the template representation for a resource type.

The returned template contains a single resource type. Each resource property is mapped to a template parameter and each resource attribute is mapped to a template output.

You can use these templates as a starting place for creating customized, template-based resources or as examples of using the particular resource in another template.

Use the `template_type` query parameter to specify the resource template type. Default type is `cfn`. The `hot` template type is supported. For example:

```
/v1/{tenant_id}/resource_types/{type_name}/template?template_type=cfn
```

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	

17.7.4.1. Request

This table shows the URI parameters for the show resource template request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{type_name}	String	The name of a resource type.

This operation does not accept a request body.

17.7.4.2. Response

Example 17.34. Show resource template: JSON response

```
{
  "HeatTemplateFormatVersion": "2012-12-12",
  "Outputs": {
    "private_key": {
      "Description": "The private key if it has been saved.",
      "Value": "{$Fn::GetAtt\": [\"KeyPair\", \"private_key\"]}"
    },
    "public_key": {
      "Description": "The public key.",
      "Value": "{$Fn::GetAtt\": [\"KeyPair\", \"public_key\"]}"
    }
  }
}
```

```

    },
    "Parameters": {
      "name": {
        "Description": "The name of the key pair.",
        "Type": "String"
      },
      "public_key": {
        "Description": "The optional public key. This allows users to
supply the public key from a pre-existing key pair. If not supplied, a new
key pair will be generated.",
        "Type": "String"
      },
      "save_private_key": {
        "AllowedValues": [
          true,
          "true",
          false,
          "false"
        ],
        "Default": false,
        "Description": "true if the system should remember a generated
private key; false otherwise.",
        "Type": "String"
      }
    },
    "Resources": {
      "KeyPair": {
        "Properties": {
          "name": {
            "Ref": "name"
          },
          "public_key": {
            "Ref": "public_key"
          },
          "save_private_key": {
            "Ref": "save_private_key"
          }
        },
        "Type": "OS::Nova::KeyPair"
      }
    }
  }
}

```

17.7.5. Show resource schema

Method	URI	Description
GET	/v1/{tenant_id}/resource_types/{type_name}	Shows the interface schema for a resource type.

A schema describes the properties that can be set on the resource, their types, constraints, descriptions, and default values. Additionally, the response shows the resource attributes and their descriptions.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	

17.7.5.1. Request

This table shows the URI parameters for the show resource schema request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{type_name}	String	The name of a resource type.

This operation does not accept a request body.

17.7.5.2. Response

Example 17.35. Show resource schema: JSON response

```
{
  "attributes": {
    "an_attribute": {
      "description": "A runtime value of the resource."
    }
  },
  "properties": {
    "a_property": {
      "constraints": [
        {
          "description": "Must be between 1 and 255 characters",
          "length": {
            "max": 255,
            "min": 1
          }
        }
      ]
    },
    "description": "A resource description.",
    "required": true,
    "type": "string",
  }
}
```

```
        "update_allowed": false
    },
    "resource_type": "OS::Heat::AResourceName",
    "support_status": {
        "message": "A status message",
        "status": "SUPPORTED",
        "version": "2014.1"
    }
}
```


17.7.6. List resource types

Method	URI	Description
GET	/v1/{tenant_id}/resource_types{?name,version,support_status}	Lists all supported template resource types.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	

17.7.6.1. Request

This table shows the URI parameters for the list resource types request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This table shows the query parameters for the list resource types request:

Name	Type	Description
name	String (Optional)	Filters the response by a resource type name. You can set this value to a regular expression. Use this filter multiple times to filter by multiple names.
version	String (Optional)	Filters the response by resource types by heat version. For example, 2016.1 or 5.0.0. Use this filter multiple times to filter by multiple versions.
support_status	String (Optional)	Filters the response by resource types by a support status. Valid support status values are UNKNOWN, SUPPORTED, DEPRECATED, UNSUPPORTED, and HIDDEN. Use this filter multiple times to filter by multiple support statuses.

This operation does not accept a request body.

17.7.6.2. Response

Example 17.36. List resource types: JSON response

```
{
  "resource_types": [
    "AWS::EC2::Instance",
    "OS::Heat::ScalingPolicy",
    "AWS::CloudFormation::Stack",
    "OS::Keystone::Group",
    "OS::Glance::Image",
    "AWS::EC2::Volume",
    "OS::Heat::SoftwareDeployment",
    "AWS::AutoScaling::ScalingPolicy",
    "AWS::EC2::InternetGateway",
```

```

"OS::Heat::SoftwareDeployments",
"AWS::EC2::VolumeAttachment",
"AWS::CloudFormation::WaitConditionHandle",
"OS::Cinder::VolumeAttachment",
"OS::Cinder::EncryptedVolumeType",
"OS::Heat::AutoScalingGroup",
"OS::Nova::FloatingIP",
"OS::Heat::HARestarter",
"OS::Keystone::Project",
"OS::Keystone::Endpoint",
"OS::Heat::InstanceGroup",
"AWS::CloudWatch::Alarm",
"AWS::AutoScaling::AutoScalingGroup",
"OS::Heat::CloudConfig",
"OS::Heat::SoftwareComponent",
"OS::Cinder::Volume",
"OS::Keystone::Service",
"OS::Heat::WaitConditionHandle",
"OS::Heat::SoftwareConfig",
"AWS::CloudFormation::WaitCondition",
"OS::Heat::StructuredDeploymentGroup",
"OS::Heat::RandomString",
"OS::Heat::SoftwareDeploymentGroup",
"OS::Nova::KeyPair",
"OS::Heat::MultipartMime",
"OS::Heat::UpdateWaitConditionHandle",
"OS::Nova::Server",
"AWS::IAM::AccessKey",
"AWS::EC2::SecurityGroup",
"AWS::EC2::EIPAssociation",
"AWS::EC2::EIP",
"OS::Heat::AccessPolicy",
"AWS::IAM::User",
"OS::Heat::WaitCondition",
"OS::Heat::StructuredDeployment",
"AWS::RDS::DBInstance",
"AWS::AutoScaling::LaunchConfiguration",
"OS::Heat::Stack",
"OS::Nova::FloatingIPAssociation",
"OS::Heat::ResourceGroup",
"OS::Heat::StructuredConfig",
"OS::Nova::ServerGroup",
"OS::Heat::StructuredDeployments",
"OS::Keystone::Role",
"OS::Keystone::User",
"AWS::ElasticLoadBalancing::LoadBalancer",
"OS::Nova::Flavor",
"OS::Cinder::VolumeType"
]
}

```

17.8. Build info

Method	URI	Description
GET	/v1/{tenant_id}/build_info	Shows build information for an Orchestration deployment.

17.8.1. Show build information

Method	URI	Description
GET	/v1/{tenant_id}/build_info	Shows build information for an Orchestration deployment.

Normal response codes: 200

17.8.1.1. Request

This table shows the URI parameters for the show build information request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

17.8.1.2. Response

Example 17.37. Show build information: JSON response

```
{
  "api": {
    "revision": "{api_build_revision}"
  },
  "engine": {
    "revision": "{engine_build_revision}"
  }
}
```

17.9. Software configuration

Method	URI	Description
POST	/v1/{tenant_id}/software_configs	Creates a software configuration.
GET	/v1/{tenant_id}/software_configs/{config_id}	Shows details for a software configuration.
DELETE	/v1/{tenant_id}/software_configs/{config_id}	Deletes a software configuration.
GET	/v1/{tenant_id}/software_deployments	Lists all available software deployments.
POST	/v1/{tenant_id}/software_deployments	Creates a software deployment.
GET	/v1/{tenant_id}/software_deployments/metadata/{server_id}	Shows the deployment configuration metadata for a server.
GET	/v1/{tenant_id}/software_deployments/{deployment_id}	Shows details for a software deployment.
PUT	/v1/{tenant_id}/software_deployments/{deployment_id}	Updates a software deployment.

Method	URI	Description
DELETE	/v1/{tenant_id}/ software_deployments/ {deployment_id}	Deletes a software deployment.

17.9.1. Create configuration

Method	URI	Description
POST	/v1/{tenant_id}/software_configs	Creates a software configuration.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	

17.9.1.1. Request

This table shows the URI parameters for the create configuration request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 17.38. Create configuration: JSON request

```
{
  "inputs": [
    {
      "default": null,
      "type": "String",
      "name": "foo",
      "description": null
    },
    {
      "default": null,
      "type": "String",
      "name": "bar",
      "description": null
    }
  ],
  "group": "script",
  "name": "a-config-we5zpvyu7b5o",
  "outputs": [
    {
      "type": "String",
      "name": "result",
      "error_output": false,
      "description": null
    }
  ],
  "config": "#!/bin/sh -x\\necho \\\"Writing to /tmp/$bar\\\"\\necho $foo > /tmp/$bar\\necho -n \\\"The file /tmp/$bar contains `cat /tmp/$bar` for server $deploy_server_id during $deploy_action\\\" > $heat_outputs_path.result\\necho \\\"Written to /tmp/$bar\\\"\\necho \\\"Output to stderr\\\" 1>&2\",
  "options": null
}
```

```
}
```

17.9.1.2. Response

Example 17.39. Create configuration: JSON response

```
{
  "software_config": {
    "creation_time": "2015-01-31T15:12:36Z",
    "inputs": [
      {
        "default": null,
        "type": "String",
        "name": "foo",
        "description": null
      },
      {
        "default": null,
        "type": "String",
        "name": "bar",
        "description": null
      }
    ],
    "group": "script",
    "name": "a-config-we5zpvyu7b5o",
    "outputs": [
      {
        "type": "String",
        "name": "result",
        "error_output": false,
        "description": null
      }
    ],
    "options": null,
    "config": "#!/bin/sh -x\\necho \\\"Writing to /tmp/$bar\\\"\\necho $foo > /tmp/$bar\\necho -n \\\"The file /tmp/$bar contains `cat /tmp/$bar` for server $deploy_server_id during $deploy_action\\\" > $heat_outputs_path.result\\necho \\\"Written to /tmp/$bar\\\"\\necho \\\"Output to stderr\\\" 1>&2\",
    "id": "ddee7aca-aa32-4335-8265-d436b20db4f1"
  }
}
```

17.9.2. Show configuration details

Method	URI	Description
GET	/v1/{tenant_id}/software_configs/{config_id}	Shows details for a software configuration.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	

17.9.2.1. Request

This table shows the URI parameters for the show configuration details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{config_id}	UUID	The UUID of the configuration.

This operation does not accept a request body.

17.9.2.2. Response

Example 17.40. Show configuration details: JSON response

```
{
  "software_config": {
    "inputs": [
      {
        "default": null,
        "type": "String",
        "name": "foo",
        "description": null
      },
      {
        "default": null,
        "type": "String",
        "name": "bar",
        "description": null
      }
    ],
    "group": "script",
    "name": "a-config-we5zpvyu7b5o",
    "outputs": [
      {
        "type": "String",
        "name": "result",
        "error_output": false,
        "description": null
      }
    ]
  }
}
```

```
    }  
  ],  
  "creation_time": "2015-01-31T15:12:36Z",  
  "id": "ddee7aca-aa32-4335-8265-d436b20db4f1",  
  "config": "#!/bin/sh -x\\necho \\\"Writing to /tmp/$bar\\\"\\necho $foo > /  
tmp/$bar\\necho -n \\\"The file /tmp/$bar contains `cat /tmp/$bar` for server  
$deploy_server_id during $deploy_action\\\" > $heat_outputs_path.result\\necho \\  
\"Written to /tmp/$bar\\\"\\necho \\\"Output to stderr\\\" 1>&2\",  
  "options": null  
}  
}
```


17.9.3. Delete config

Method	URI	Description
DELETE	/v1/{tenant_id}/software_configs/{config_id}	Deletes a software configuration.

Normal response codes: 204

17.9.3.1. Request

This table shows the URI parameters for the delete config request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{config_id}	UUID	The UUID of the configuration.

This operation does not accept a request body.

17.9.4. List deployments

Method	URI	Description
GET	/v1/{tenant_id}/software_deployments	Lists all available software deployments.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400	Bad Request	
401	Unauthorized	
404	Not Found	
500	Internal Server Error	

17.9.4.1. Request

This table shows the URI parameters for the list deployments request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

17.9.4.2. Response

Example 17.41. List deployments: JSON response

```
{
  "software_deployments": [
    {
      "status": "COMPLETE",
      "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
      "config_id": "8da95794-2ad9-4979-8ae5-739ce314c5cd",
      "output_values": {
        "deploy_stdout": "Writing to /tmp/barmy\nWritten to /tmp/barmy\n",
        "deploy_stderr": "+ echo Writing to /tmp/barmy\n+ echo fu\n\n+ cat /tmp/barmy\n+ echo -n The file /tmp/barmy contains fu for server\nec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE\n+ echo Written to /tmp/barmy\n+ echo Output to stderr\nOutput to stderr\n",
        "deploy_status_code": 0,
        "result": "The file /tmp/barmy contains fu for server\nec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE"
      },
      "input_values": null,
      "action": "CREATE",
      "status_reason": "Outputs received",
      "id": "ef422fa5-719a-419e-a10c-72e3a367b0b8",
      "creation_time": "2015-01-31T15:12:36Z",
      "updated_time": "2015-01-31T15:18:21Z"
    }
  ]
}
```

```
    ]  
}
```

17.9.5. Create deployment

Method	URI	Description
POST	/v1/{tenant_id}/software_deployments	Creates a software deployment.

Normal response codes: 200

17.9.5.1. Request

This table shows the URI parameters for the create deployment request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 17.42. Create deployment: JSON request

```
{
  "status": "IN_PROGRESS",
  "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
  "config_id": "8da95794-2ad9-4979-8ae5-739ce314c5cd",
  "stack_user_project_id": "c024bfada67845ddb17d2b0c0be8cd79",
  "action": "CREATE",
  "status_reason": "Deploy data available"
}
```

17.9.5.2. Response

Example 17.43. Create deployment: JSON response

```
{
  "software_deployment": {
    "status": "IN_PROGRESS",
    "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
    "config_id": "8da95794-2ad9-4979-8ae5-739ce314c5cd",
    "output_values": null,
    "input_values": null,
    "action": "CREATE",
    "status_reason": "Deploy data available",
    "id": "ef422fa5-719a-419e-a10c-72e3a367b0b8",
    "creation_time": "2015-01-31T15:12:36Z",
    "updated_time": "2015-01-31T15:18:21Z"
  }
}
```

17.9.6. Show server configuration metadata

Method	URI	Description
GET	/v1/{tenant_id}/software_deployments/metadata/{server_id}	Shows the deployment configuration metadata for a server.

Use the group property to specify the configuration hook to which the pass the metadata item.

Normal response codes: 200

17.9.6.1. Request

This table shows the URI parameters for the show server configuration metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{server_id}	UUID	The UUID of the server for which to fetch configuration metadata.

This operation does not accept a request body.

17.9.6.2. Response

Example 17.44. Show server configuration metadata: JSON response

```
{
  "metadata": [
    {
      "inputs": [
        {
          "default": null,
          "type": "String",
          "name": "foo",
          "value": "fooooo",
          "description": null
        },
        {
          "default": null,
          "type": "String",
          "name": "bar",
          "value": "baaaaa",
          "description": null
        },
        {
          "type": "String",
          "name": "deploy_server_id",
          "value": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
          "description": "ID of the server being deployed to"
        },
        {
          "type": "String",
          "name": "deploy_action",
          "value": "CREATE",
          "description": "Name of the current action being deployed"
        }
      ]
    }
  ]
}
```

```

        {
            "type": "String",
            "name": "deploy_stack_id",
            "value": "a/9bd57090-8954-48ab-bab9-adf9e1ac70fc",
            "description": "ID of the stack this deployment belongs
to"
        },
        {
            "type": "String",
            "name": "deploy_resource_name",
            "value": "deployment",
            "description": "Name of this deployment resource in the
stack"
        },
        {
            "type": "String",
            "name": "deploy_signal_id",
            "value": "http://192.168.20.103:8000/v1/signal/arn
%3Aopenstack%3Aheat%3A%3Ae2a84fbdaeb047ae8da4b503f3b69f1f%3Astacks%2Fa
%2F9bd57090-8954-48ab-bab9-adf9e1ac70fc%2Fresources%2Fdeployment?Timestamp=
2014-03-19T20%3A30%3A59Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=
ca3571413e4a49998d580215517b3685&SignatureVersion=2&Signature=w6IU
%2BNbg86mqwSOUf1GLuKP07KaD82PiGpL4ig9Q1l4%3D",
            "description": "ID of signal to use for signalling output
values"
        }
    ],
    "group": "script",
    "name": "a-config-we5zpvyu7b5o",
    "outputs": [
        {
            "type": "String",
            "name": "result",
            "error_output": false,
            "description": null
        }
    ],
    "options": null,
    "creation_time": "2015-01-31T15:12:36Z",
    "updated_time": "2015-01-31T15:18:21Z",
    "config": "#!/bin/sh -x\nnecho \"Writing to /tmp/$bar\"\\necho $foo
> /tmp/$bar\\necho -n \"The file /tmp/$bar contains `cat /tmp/$bar` for server
$deploy_server_id during $deploy_action\" > $heat_outputs_path.result\\necho \\
\"Written to /tmp/$bar\"\\necho \"Output to stderr\" 1>&2",
    "id": "3d5ec2a8-7004-43b6-a7f6-542bdbbe9d434"
},
{
    "inputs": [
        {
            "default": null,
            "type": "String",
            "name": "foo",
            "value": "fu",
            "description": null
        },
        {
            "default": null,
            "type": "String",
            "name": "bar",
            "value": "barmy",

```

```

        "description": null
    },
    {
        "type": "String",
        "name": "deploy_server_id",
        "value": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
        "description": "ID of the server being deployed to"
    },
    {
        "type": "String",
        "name": "deploy_action",
        "value": "CREATE",
        "description": "Name of the current action being deployed"
    },
    {
        "type": "String",
        "name": "deploy_stack_id",
        "value": "a/9bd57090-8954-48ab-bab9-adf9e1ac70fc",
        "description": "ID of the stack this deployment belongs
to"
    },
    {
        "type": "String",
        "name": "deploy_resource_name",
        "value": "other_deployment",
        "description": "Name of this deployment resource in the
stack"
    },
    {
        "type": "String",
        "name": "deploy_signal_id",
        "value": "http://192.168.20.103:8000/v1/signal/arn
%3Aopenstack%3Aheat%3A%3Ae2a84fbdaeb047ae8da4b503f3b69f1f%3Astacks%2Fa
%2F9bd57090-8954-48ab-bab9-adf9e1ac70fc%2Fresources%2Fother_deployment?
Timestamp=2014-03-19T20%3A30%3A59Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=
7b761482f8254946bcd3d5ccb36fe939&SignatureVersion=2&Signature=giMfv%2BhrAw6y
%2FCMKQIQz2Ih05PkAj5%2BFp5YsL6rul3o%3D",
        "description": "ID of signal to use for signalling output
values"
    }
],
"group": "script",
"name": "a-config-we5zpvyyu7b5o",
"outputs": [
    {
        "type": "String",
        "name": "result",
        "error_output": false,
        "description": null
    }
],
"options": null,
"creation_time": "2015-01-31T16:14:13Z",
"updated_time": "2015-01-31T16:18:19Z",
"config": "#!/bin/sh -x\\necho \\\"Writing to /tmp/$bar\\\"\\necho $foo
> /tmp/$bar\\necho -n \\\"The file /tmp/$bar contains `cat /tmp/$bar` for server
$deploy_server_id during $deploy_action\\\" > $heat_outputs_path.result\\necho \\
\"Written to /tmp/$bar\\\"\\necho \\\"Output to stderr\\\" 1>&2\",
"id": "8da95794-2ad9-4979-8ae5-739ce314c5cd"
}

```

```
    ]  
}
```


17.9.7. Show deployment details

Method	URI	Description
GET	/v1/{tenant_id}/software_deployments/{deployment_id}	Shows details for a software deployment.

Normal response codes: 200

17.9.7.1. Request

This table shows the URI parameters for the show deployment details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{deployment_id}	UUID	The UUID of the deployment.

This operation does not accept a request body.

17.9.7.2. Response

Example 17.45. Show deployment details: JSON response

```
{
  "software_deployment": {
    "status": "IN_PROGRESS",
    "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
    "config_id": "3d5ec2a8-7004-43b6-a7f6-542bdbe9d434",
    "output_values": null,
    "input_values": null,
    "action": "CREATE",
    "status_reason": "Deploy data available",
    "id": "06e87bcc-33a2-4bce-aebd-533e698282d3",
    "creation_time": "2015-01-31T15:12:36Z",
    "updated_time": "2015-01-31T15:18:21Z"
  }
}
```

17.9.8. Update deployment

Method	URI	Description
PUT	/v1/{tenant_id}/software_deployments/{deployment_id}	Updates a software deployment.

Normal response codes: 200

17.9.8.1. Request

This table shows the URI parameters for the update deployment request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{deployment_id}	UUID	The UUID of the deployment.

Example 17.46. Update deployment: JSON request

```
{
  "status": "COMPLETE",
  "output_values": {
    "deploy_stdout": "Writing to /tmp/baaaaa\nWritten to /tmp/baaaaa\n",
    "deploy_stderr": "+ echo Writing to /tmp/baaaaa\n+ echo fooooo\n+
cat /tmp/baaaaa\n+ echo -n The file /tmp/baaaaa contains fooooo for server
ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE\n+ echo Written to /tmp/
baaaaa\n+ echo Output to stderr\nOutput to stderr\n",
    "deploy_status_code": 0,
    "result": "The file /tmp/baaaaa contains fooooo for server
ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE"
  },
  "status_reason": "Outputs received"
}
```

17.9.8.2. Response

Example 17.47. Update deployment: JSON response

```
{
  "software_deployment": {
    "status": "COMPLETE",
    "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
    "config_id": "3d5ec2a8-7004-43b6-a7f6-542bde9d434",
    "output_values": {
      "deploy_stdout": "Writing to /tmp/baaaaa\nWritten to /tmp/baaaaa\
n",
      "deploy_stderr": "+ echo Writing to /tmp/baaaaa\n+ echo fooooo\n
+ cat /tmp/baaaaa\n+ echo -n The file /tmp/baaaaa contains fooooo for server
ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE\n+ echo Written to /tmp/
baaaaa\n+ echo Output to stderr\nOutput to stderr\n",
      "deploy_status_code": 0,
      "result": "The file /tmp/baaaaa contains fooooo for server
ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE"
    },
  },
}
```

```
    "input_values": null,  
    "action": "CREATE",  
    "status_reason": "Outputs received",  
    "id": "06e87bcc-33a2-4bce-aebd-533e698282d3",  
    "creation_time": "2015-01-31T15:12:36Z",  
    "updated_time": "2015-01-31T15:18:21Z"  
  }  
}
```

17.9.9. Delete deployment

Method	URI	Description
DELETE	/v1/{tenant_id}/software_deployments/{deployment_id}	Deletes a software deployment.

Normal response codes: 204

17.9.9.1. Request

This table shows the URI parameters for the delete deployment request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.
{deployment_id}	UUID	The UUID of the deployment.

This operation does not accept a request body.

17.10. Manage service

Method	URI	Description
GET	/v1/{tenant_id}/services	Enables administrative users to view details for all orchestration engines.

17.10.1. Show orchestration engine status

Method	URI	Description
GET	/v1/{tenant_id}/services	Enables administrative users to view details for all orchestration engines.

Orchestration engine details include `engine_id`, topic name, last updated time, health status, and host name.

Troubleshooting

- A 503 error code indicates that the heat engines are not operational. Run the **heat-manage service list** command or contact your cloud provider to determine why the heat engines are not operational.

Normal response codes: 200

Error response codes: 503, 403

17.10.1.1. Request

This table shows the URI parameters for the show orchestration engine status request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

17.10.1.2. Response

Example 17.48. Show orchestration engine status: JSON response

```
{
  "services": [
    {
      "status": "up",
      "binary": "heat-engine",
      "report_interval": 60,
      "engine_id": "9d9242c3-4b9e-45e1-9e74-7615fbf20e5d",
      "created_at": "2015-02-03T05:55:59.000000",
      "hostname": "mrkanag",
      "updated_at": "2015-02-03T05:57:59.000000",
      "topic": "engine",
      "host": "engine-1",
      "deleted_at": null,
      "id": "e1908f44-42f9-483f-b778-bc814072c33d"
    },
    {
      "status": "down",
      "binary": "heat-engine",
      "report_interval": 60,
      "engine_id": "2d2434bf-adb6-4453-9c6b-b22fb8bd2306",
      "created_at": "2015-02-03T06:03:14.000000",
```

```
        "hostname": "mrkanag",
        "updated_at": "2015-02-03T06:09:55.000000",
        "topic": "engine",
        "host": "engine",
        "deleted_at": null,
        "id": "582b5657-6db7-48ad-8483-0096350faa21"
    }
]
```

18. Shared File Systems API v2 (CURRENT)

Provides coordinated access to shared or distributed file systems.

The Shared File Systems API uses Compute-style micro-versions. Use the HTTP X-Openstack-Manila-API-Version request header to specify a valid micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

Other than the switch to the micro-versions approach, the Shared File Systems API v2.0 is functionally identical to the [Shared File Systems API v1](#).

Subsequent API v2 micro-versions, such as v2.1, differ from API v1. Look for notes that identify in which post-v2 micro-version a feature, method, or parameter was introduced.

Latest micro-version for Liberty release of Manila is 2.6

Latest micro-version for Mitaka release of Manila is 2.15

[History of all Manila API changes.](#)

Method	URI	Description
API extensions		
GET	/v2/{tenant_id}/extensions	Lists all extensions.
Limits		
GET	/v2/{tenant_id}/limits	Lists share limits.
Shares		
POST	/v2/{tenant_id}/shares	Creates a share.
GET	/v2/{tenant_id}/shares{?all_tenants,name,status,share_server_id,metadata,extra_specs,share_type_id,limit,offset,sort_key,sort_dir,snapshot_id,host,share_network_id,project_id,consistency_group_id}	Lists all shares.
GET	/v2/{tenant_id}/shares/detail{?all_tenants,name,status,share_server_id,metadata,extra_specs,share_type_id,limit,offset,sort_key,sort_dir,snapshot_id,host,share_network_id,project_id,consistency_group_id}	Lists all shares, with details.
POST	/v2/{tenant_id}/shares/manage	Configures Shared File Systems to manage a share. This API is available for micro-versions later than or equal to 2.7
GET	/v2/{tenant_id}/shares/{share_id}	Shows details for a share.
PUT	/v2/{tenant_id}/shares/{share_id}	Updates a share.
DELETE	/v2/{tenant_id}/shares/{share_id}{?consistency_group_id}	Deletes a share.
Share export locations (since API v2.9)		
GET	/v2/{tenant_id}/shares/{share_id}/export_locations	Lists all export locations for a share.

Method	URI	Description
GET	/v2/{tenant_id}/shares/{share_id}/export_locations/{export_location_id}	Show details of an export location belonging to a share.
Share metadata		
GET	/v2/{tenant_id}/shares/{share_id}/metadata	Shows the metadata for a share.
PUT	/v2/{tenant_id}/shares/{share_id}/metadata	Updates the metadata for a share.
POST	/v2/{tenant_id}/shares/{share_id}/metadata	Sets the metadata on a share.
DELETE	/v2/{tenant_id}/shares/{share_id}/metadata/{key}	Unsets the metadata on a share.
Share actions		
POST	/v2/{tenant_id}/shares/{share_id}/action	Grants access to a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Revokes access from a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Lists access rules for a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Administrator only. Explicitly updates the state of a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Administrator only. Force-deletes a share in any state.
POST	/v2/{tenant_id}/shares/{share_id}/action	Increases the size of a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Shrinks the size of a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Configures Shared File Systems to stop managing a share.
Share snapshots		
POST	/v2/{tenant_id}/snapshots	Creates a snapshot from a share.
GET	/v2/{tenant_id}/snapshots	Lists all share snapshots.
POST	/v2/{tenant_id}/snapshots/manage	(Since API v2.12) Configures Shared File Systems to manage a share snapshot.
GET	/v2/{tenant_id}/snapshots/detail	Lists all share snapshots with details.
GET	/v2/{tenant_id}/snapshots/{snapshot_id}	Shows details for a share snapshot.
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}	Updates a share snapshot.
DELETE	/v2/{tenant_id}/snapshots/{snapshot_id}	Deletes a share snapshot.
POST	/v2/{tenant_id}/snapshots/{snapshot_id}/action	(Since API v2.12) Configures Shared File Systems to stop managing a share snapshot.
POST	/v2/{tenant_id}/snapshots/{snapshot_id}/action	Administrator only. Explicitly updates the state of a share snapshot.
POST	/v2/{tenant_id}/snapshots/{snapshot_id}/action	Administrator only. Force-deletes a share snapshot in any state.
Share networks		
POST	/v2/{tenant_id}/share-networks	Creates a share network.
GET	/v2/{tenant_id}/share-networks	Lists all share networks.

Method	URI	Description
GET	/v2/{tenant_id}/share-networks/detail	Lists all share networks with details.
GET	/v2/{tenant_id}/share-networks/{share_network_id}	Shows details for a share network.
PUT	/v2/{tenant_id}/share-networks/{share_network_id}	Updates a share network.
DELETE	/v2/{tenant_id}/share-networks/{share_network_id}	Deletes a share network.
POST	/v2/{tenant_id}/share-networks/{share_network_id}/action	Adds a security service to a share network.
POST	/v2/{tenant_id}/share-networks/{share_network_id}/action	Removes a security service from a share network.
Security services		
POST	/v2/{tenant_id}/security-services	Creates a security service.
GET	/v2/{tenant_id}/security-services	Lists all security services.
GET	/v2/{tenant_id}/security-services/detail	Lists all security services with details.
GET	/v2/{tenant_id}/security-services/{security_service_id}	Shows details for a security service.
PUT	/v2/{tenant_id}/security-services/{security_service_id}	Updates a security service.
DELETE	/v2/{tenant_id}/security-services/{security_service_id}	Deletes a security service.
Share servers		
GET	/v2/{tenant_id}/share-servers	Lists all share servers.
GET	/v2/{tenant_id}/share-servers/{share_server_id}	Shows details for a share server.
DELETE	/v2/{tenant_id}/share-servers/{share_server_id}	Deletes a share server.
Share instances (since API v2.3)		
GET	/v2/{tenant_id}/share_instances	Lists all share instances.
GET	/v2/{tenant_id}/share_instances/{share_instance_id}	Shows details for a share instance.
POST	/v2/{tenant_id}/share_instances/{share_instance_id}/action	Administrator only. Explicitly updates the state of a share instance.
POST	/v2/{tenant_id}/share_instances/{share_instance_id}/action	Administrator only. Force-deletes a share instance.
Share instance export locations (since API v2.9)		
GET	/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations	Lists all export locations for a share instance.
GET	/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations/{export_location_id}	Show details of an export location belonging to a share instance.
Share types		
POST	/v2/{tenant_id}/types	Creates a share type.
GET	/v2/{tenant_id}/types	Lists all share types.
GET	/v2/{tenant_id}/types/default	Lists default share types.
DELETE	/v2/{tenant_id}/types/{share_type_id}	Deletes a share type.

Method	URI	Description
GET	/v2/{tenant_id}/types/{share_type_id}/extra_specs	Lists the extra specifications for a share type.
POST	/v2/{tenant_id}/types/{share_type_id}/extra_specs	Sets an extra specification for the share type.
DELETE	/v2/{tenant_id}/types/{share_type_id}/extra_specs/{key}	Unsets an extra specification for the share type.
POST	/v2/{tenant_id}/types/{share_type_id}/action	Adds share type access for a project.
POST	/v2/{tenant_id}/types/{share_type_id}/action	Removes share type access from a project.
GET	/v2/{tenant_id}/types/{share_type_id}/share_type_access	Shows access details for a share type.
Back-end storage pools		
GET	/v2/{tenant_id}/scheduler-stats/pools	Lists all back-end storage pools.
GET	/v2/{tenant_id}/scheduler-stats/pools/detail	Lists all storage pools for a back end, with details.
Services		
GET	/v2/{tenant_id}/services	Lists all services.
PUT	/v2/{tenant_id}/services/enable	Enables a service.
PUT	/v2/{tenant_id}/services/disable	Disables a service.
Availability zones		
GET	/v2/{tenant_id}/availability-zones	Lists all availability zones.
Manage share		
POST	/v2/{tenant_id}/os-share-manage	Configures Shared File Systems to manage a share.
POST	/v2/{tenant_id}/os-share-unmanage/{share_id}/unmanage	Configures Shared File Systems to stop managing a share.
Quota sets		
GET	/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}	Shows quotas for a tenant.
PUT	/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}	Updates quotas for a tenant.
DELETE	/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}	Deletes quotas for a tenant. The quota reverts to the default quota.
GET	/v2/{tenant_id}/quota-sets/{tenant_id}/defaults	Shows default quotas for a tenant.

18.1. API extensions

Lists available Shared File Systems API extensions.

Method	URI	Description
GET	/v2/{tenant_id}/extensions	Lists all extensions.

18.1.1. List extensions

Method	URI	Description
GET	/v2/{tenant_id}/extensions	Lists all extensions.

Normal response codes: 200

18.1.1.1. Request

This table shows the header parameters for the list extensions request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list extensions request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.1.1.2. Response

Example 18.1. List extensions: JSON response

```
{
  "extensions": [
    {
      "alias": "os-extended-quotas",
      "updated": "2013-06-09T00:00:00+00:00",
      "name": "ExtendedQuotas",
      "links": [],
      "description": "Extend quotas. Adds ability for admins to delete
quota and optionally force the update Quota command."
    },
    {
      "alias": "os-quota-sets",
      "updated": "2011-08-08T00:00:00+00:00",
      "name": "Quotas",
      "links": [],
      "description": "Quotas management support."
    },
    {
      "alias": "os-quota-class-sets",
      "updated": "2012-03-12T00:00:00+00:00",
      "name": "QuotaClasses",
      "links": [],
      "description": "Quota classes management support."
    },
    {
      "alias": "os-share-unmanage",
      "updated": "2015-02-17T00:00:00+00:00",
      "name": "ShareUnmanage",
      "links": [],
      "description": "Enable share unmanage operation."
    }
  ]
}
```

```

    },
    {
      "alias": "os-types-manage",
      "updated": "2011-08-24T00:00:00+00:00",
      "name": "TypesManage",
      "links": [],
      "description": "Types manage support."
    },
    {
      "alias": "share-actions",
      "updated": "2012-08-14T00:00:00+00:00",
      "name": "ShareActions",
      "links": [],
      "description": "Enable share actions."
    },
    {
      "alias": "os-availability-zone",
      "updated": "2015-07-28T00:00:00+00:00",
      "name": "AvailabilityZones",
      "links": [],
      "description": "Describe Availability Zones."
    },
    {
      "alias": "os-user-quotas",
      "updated": "2013-07-18T00:00:00+00:00",
      "name": "UserQuotas",
      "links": [],
      "description": "Project user quota support."
    },
    {
      "alias": "os-share-type-access",
      "updated": "2015-03-02T00:00:00Z",
      "name": "ShareTypeAccess",
      "links": [],
      "description": "share type access support."
    },
    {
      "alias": "os-types-extra-specs",
      "updated": "2011-08-24T00:00:00+00:00",
      "name": "TypesExtraSpecs",
      "links": [],
      "description": "Type extra specs support."
    },
    {
      "alias": "os-admin-actions",
      "updated": "2015-08-03T00:00:00+00:00",
      "name": "AdminActions",
      "links": [],
      "description": "Enable admin actions."
    },
    {
      "alias": "os-used-limits",
      "updated": "2014-03-27T00:00:00+00:00",
      "name": "UsedLimits",
      "links": [],
      "description": "Provide data on limited resources that are being
used."
    },
    {
      "alias": "os-services",

```

```
    "updated": "2012-10-28T00:00:00-00:00",
    "name": "Services",
    "links": [],
    "description": "Services support."
  },
  {
    "alias": "os-share-manage",
    "updated": "2015-02-17T00:00:00+00:00",
    "name": "ShareManage",
    "links": [],
    "description": "Allows existing share to be 'managed' by Manila."
  }
]
```

18.2. Limits

Limits are the resource limitations that are allowed for each tenant (project). An administrator can configure limits in the `manila.conf` file.

Users can query their rate and absolute limits. The absolute limits contain information about:

- Total maximum share memory, in GBs.
- Number of share-networks.
- Number of share-snapshots.
- Number of shares.
- Shares and total used memory, in GBs.
- Snapshots and total used memory, in GBs.

Rate limits control the frequency at which users can issue specific API requests. Administrators use rate limiting to configure limits on the type and number of API calls that can be made in a specific time interval. For example, a rate limit can control the number of **GET** requests that can be processed during a one-minute period.

Method	URI	Description
GET	<code>/v2/{tenant_id}/limits</code>	Lists share limits.

18.2.1. List share limits

Method	URI	Description
GET	/v2/{tenant_id}/limits	Lists share limits.

Normal response codes: 200

18.2.1.1. Request

This table shows the header parameters for the list share limits request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list share limits request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.2.1.2. Response

Example 18.2. List share limits: JSON response

```
{
  "limits": {
    "rate": [],
    "absolute": {
      "totalShareNetworksUsed": 0,
      "maxTotalShareGigabytes": 1000,
      "maxTotalShareNetworks": 10,
      "totalSharesUsed": 0,
      "totalShareGigabytesUsed": 0,
      "totalShareSnapshotsUsed": 0,
      "maxTotalShares": 50,
      "totalSnapshotGigabytesUsed": 0,
      "maxTotalSnapshotGigabytes": 1000,
      "maxTotalShareSnapshots": 50
    }
  }
}
```

18.3. Shares

A share is a remote, mountable file system. You can mount a share to and access a share from several hosts by several users at a time.

You can create a share and associate it with a network, list shares, and show information for, update, and delete a share.

To create a share, specify one of these supported protocols:

- NFS. Network File System (NFS).
- CIFS. Common Internet File System (CIFS).
- GLUSTERFS. Gluster file system (GlusterFS).
- HDFS. Hadoop Distributed File System (HDFS).
- CEPHFS. Ceph File System (CephFS).

You can also create snapshots of shares. To create a snapshot, you specify the ID of the share that you want to snapshot.

A share has one of these status values:

Table 18.1. Share statuses

Status	Description
creating	The share is being created.
deleting	The share is being deleted.
error	An error occurred during share creation.
error_deleting	An error occurred during share deletion.
available	The share is ready to use.
manage_starting	Share manage started.
manage_error	Share manage failed.
unmanage_starting	Share unmanage started.
unmanage_error	Share cannot be unmanaged.
unmanaged	Share was unmanaged.
extending	The extend, or increase, share size request was issued successfully.
extending_error	Extend share failed.
shrinking	Share is being shrunk.
shrinking_error	Failed to update quota on share shrinking.
shrinking_possible_data_loss	Shrink share failed due to possible data loss.

Method	URI	Description
POST	/v2/{tenant_id}/shares	Creates a share.
GET	/v2/{tenant_id}/shares{?all_tenants,name,status,share_server_id,metadata,extra_specs,share_type_id,limit,offset,sort_key,sort_dir,snapshot_id,host,share_network_id,project_id,consistency_group_id}	Lists all shares.
GET	/v2/{tenant_id}/shares/detail{?all_tenants,name,status,share_server_id,metadata,extra_specs,share_type_id,limit,offset,sort_key,sort_dir,snapshot_id,host,share_network_id,project_id,consistency_group_id}	Lists all shares, with details.

Method	URI	Description
POST	/v2/{tenant_id}/shares/manage	Configures Shared File Systems to manage a share. This API is available for micro-versions later than or equal to 2.7
GET	/v2/{tenant_id}/shares/{share_id}	Shows details for a share.
PUT	/v2/{tenant_id}/shares/{share_id}	Updates a share.
DELETE	/v2/{tenant_id}/shares/{share_id} {?consistency_group_id}	Deletes a share.

18.3.1. Create share

Method	URI	Description
POST	/v2/{tenant_id}/shares	Creates a share.

Normal response codes: 200

18.3.1.1. Request

This table shows the header parameters for the create share request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the create share request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 18.3. Create share: JSON request

```
{
  "share": {
    "description": "My custom share London",
    "share_type": null,
    "share_proto": "nfs",
    "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
    "name": "share_London",
    "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
    "snapshot_id": null,
    "is_public": true,
    "size": 1,
    "metadata": {
      "project": "my_app",
      "aim": "doc"
    }
  }
}
```

18.3.1.2. Response

Example 18.4. Create share: JSON response

```
{
  "share": {
    "status": null,
    "share_server_id": null,
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "name": "share_London",
    "share_type": "25747776-08e5-494f-ab40-a64b9d20d8f7",
    "share_type_name": "default",
    "availability_zone": null,
  }
}
```

```

    "created_at": "2015-09-18T10:25:24.533287",
    "export_location": null,
    "links": [
      {
        "href": "http://172.18.198.54:8786/v1/
16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
        "rel": "self"
      },
      {
        "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
        "rel": "bookmark"
      }
    ],
    "share_network_id": null,
    "export_locations": [],
    "share_proto": "NFS",
    "host": null,
    "access_rules_status": "active",
    "has_replicas": false,
    "replication_type": null,
    "task_state": null,
    "snapshot_support": true,
    "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
    "source_cgsnapshot_member_id": null,
    "volume_type": "default",
    "snapshot_id": null,
    "is_public": true,
    "metadata": {
      "project": "my_app",
      "aim": "doc"
    },
    "id": "011d21e2-fbc3-4e4a-9993-9ea223f73264",
    "size": 1,
    "description": "My custom share London"
  }
}

```

18.3.2. List shares

Method	URI	Description
GET	/v2/{tenant_id}/shares{? all_tenants,name,status, share_server_id,metadata, extra_specs,share_type_id,limit,offset,sort_key,sort_dir, snapshot_id,host,share_network_id, project_id,consistency_group_id}	Lists all shares.

Normal response codes: 200

18.3.2.1. Request

This table shows the header parameters for the list shares request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list shares request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list shares request:

Name	Type	Description
all_tenants	Boolean (Optional)	(Admin only). Defines whether to list shares for all tenants. Set to 1 to list shares for all tenants. Set to 0 to list shares only for the current tenant.
name	String (Optional)	The share name.
status	String (Optional)	Filters by a share status. A valid value is creating, error, available, deleting, error_deleting, manage_starting, manage_error, unmanage_starting, unmanage_error, unmanaged, extending, extending_error, shrinking, shrinking_error, or shrinking_possible_data_loss_error.
share_server_id	UUID (Optional)	The UUID of the share server.
metadata	Dict (Optional)	One or more metadata key-value pairs, as a dictionary of strings.
extra_specs	String (Optional)	The extra specifications as a set of one or more key-value pairs. In each pair, the key is the name of the extra specification and the value is the share type that was used to create the share.
share_type_id	UUID (Optional)	(Since API v2.6) The UUID of the share type.

Name	Type	Description
limit	Int (Optional)	The maximum number of shares to return.
offset	Int (Optional)	The offset to define start point of share listing.
sort_key	String (Optional)	The key to sort a list of shares. A valid value is id, status, size, host, share_proto, export_location, availability_zone, user_id, project_id, created_at, updated_at, display_name, name, share_type_id, share_type, share_network_id, share_network, snapshot_id, or snapshot.
sort_dir	String (Optional)	The direction to sort a list of shares. A valid value is asc, or desc.
snapshot_id	UUID (Optional)	The UUID of the snapshot that was used to create the share.
host	String (Optional)	The share host name.
share_network_id	UUID (Optional)	The UUID of the share network.
project_id	UUID (Optional)	The UUID of the project in which the share was created. Useful with all_tenants parameter.
consistency_group_id	UUID (Optional)	(Since API v2.4) The UUID of the consistency group where the share was created.

This operation does not accept a request body.

18.3.2.2. Response

Example 18.5. List shares: JSON response

```
{
  "shares": [
    {
      "id": "d94a8548-2079-4be0-b21c-0a887acd31ca",
      "links": [
        {
          "href": "http://172.18.198.54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/shares/d94a8548-2079-4be0-b21c-0a887acd31ca",
          "rel": "self"
        },
        {
          "href": "http://172.18.198.54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/shares/d94a8548-2079-4be0-b21c-0a887acd31ca",
          "rel": "bookmark"
        }
      ],
      "name": "My_share"
    },
    {
      "id": "406ea93b-32e9-4907-a117-148b3945749f",
      "links": [

```

```
        "href": "http://172.18.198.54:8786/v1/
16e1ab15c35a457e9c2b2aa189f544e1/shares/406ea93b-32e9-4907-a117-148b3945749f",
        "rel": "self"
    },
    {
        "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/406ea93b-32e9-4907-a117-148b3945749f",
        "rel": "bookmark"
    }
],
"name": "Share1"
}
]
```

18.3.3. List shares with details

Method	URI	Description
GET	/v2/{tenant_id}/shares/detail{?all_tenants,name,status,share_server_id,metadata,extra_specs,share_type_id,limit,offset,sort_key,sort_dir,snapshot_id,host,share_network_id,project_id,consistency_group_id}	Lists all shares, with details.

Normal response codes: 200

18.3.3.1. Request

This table shows the header parameters for the list shares with details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list shares with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list shares with details request:

Name	Type	Description
all_tenants	Boolean (Optional)	(Admin only). Defines whether to list shares for all tenants. Set to 1 to list shares for all tenants. Set to 0 to list shares only for the current tenant.
name	String (Optional)	The share name.
status	String (Optional)	Filters by a share status. A valid value is creating, error, available, deleting, error_deleting, manage_starting, manage_error, unmanage_starting, unmanage_error, unmanaged, extending, extending_error, shrinking, shrinking_error, or shrinking_possible_data_loss_error.
share_server_id	UUID (Optional)	The UUID of the share server.
metadata	Dict (Optional)	One or more metadata key-value pairs, as a dictionary of strings.
extra_specs	String (Optional)	The extra specifications as a set of one or more key-value pairs. In each pair, the key is the name of the extra specification and the value is the share type that was used to create the share.
share_type_id	UUID (Optional)	(Since API v2.6) The UUID of the share type.
limit	Int (Optional)	The maximum number of shares to return.
offset	Int	The offset to define start point of share listing.

Name	Type	Description
	<i>(Optional)</i>	
sort_key	String <i>(Optional)</i>	The key to sort a list of shares. A valid value is id, status, size, host, share_proto, export_location, availability_zone, user_id, project_id, created_at, updated_at, display_name, name, share_type_id, share_type, share_network_id, share_network, snapshot_id, or snapshot.
sort_dir	String <i>(Optional)</i>	The direction to sort a list of shares. A valid value is asc, or desc.
snapshot_id	UUID <i>(Optional)</i>	The UUID of the snapshot that was used to create the share.
host	String <i>(Optional)</i>	The share host name.
share_network_id	UUID <i>(Optional)</i>	The UUID of the share network.
project_id	UUID <i>(Optional)</i>	The UUID of the project in which the share was created. Useful with all_tenants parameter.
consistency_group_id	UUID <i>(Optional)</i>	(Since API v2.4) The UUID of the consistency group where the share was created.

This operation does not accept a request body.

18.3.3.2. Response

Example 18.6. List shares with details: JSON response

```
{
  "shares": [
    {
      "links": [
        {
          "href": "http://172.18.198.54:8786/v2/16e1ab15c35a457e9c2b2aa189f544e1/shares/f45cc5b2-d1bb-4a3e-ba5b-5c4125613adc",
          "rel": "self"
        },
        {
          "href": "http://172.18.198.54:8786/16e1ab15c35a457e9c2b2aa189f544e1/shares/f45cc5b2-d1bb-4a3e-ba5b-5c4125613adc",
          "rel": "bookmark"
        }
      ],
      "availability_zone": "nova",
      "share_network_id": "f9b2e754-ac01-4466-86e1-5c569424754e",
      "export_locations": [],
      "share_server_id": "87d8943a-f5da-47a4-b2f2-ddfa6794aa82",
      "snapshot_id": null,
      "id": "f45cc5b2-d1bb-4a3e-ba5b-5c4125613adc",
      "size": 1,
      "share_type": "25747776-08e5-494f-ab40-a64b9d20d8f7",
      "share_type_name": "default",
      "export_location": null,
      "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
      "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    }
  ]
}
```

```

    "metadata": {},
    "status": "error",
    "access_rules_status": "active",
    "description": "There is a share description.",
    "host": "manila2@generic1#GENERIC1",
    "task_state": null,
    "is_public": true,
    "snapshot_support": true,
    "name": "my_share4",
    "has_replicas": false,
    "replication_type": null,
    "created_at": "2015-09-16T18:19:50.000000",
    "share_proto": "NFS",
    "volume_type": "default",
    "source_cgsnapshot_member_id": null
  },
  {
    "links": [
      {
        "href": "http://172.18.198.54:8786/v2/16e1ab15c35a457e9c2b2aa189f544e1/shares/c4a2ced4-2c9f-4ae1-adaa-6171833e64df",
        "rel": "self"
      },
      {
        "href": "http://172.18.198.54:8786/16e1ab15c35a457e9c2b2aa189f544e1/shares/c4a2ced4-2c9f-4ae1-adaa-6171833e64df",
        "rel": "bookmark"
      }
    ],
    "availability_zone": "nova",
    "share_network_id": "f9b2e754-ac01-4466-86e1-5c569424754e",
    "export_locations": [
      "10.254.0.5:/shares/share-50ad5e7b-f6f1-4b78-a651-0812cef2bb67"
    ],
    "share_server_id": "87d8943a-f5da-47a4-b2f2-ddfa6794aa82",
    "snapshot_id": null,
    "id": "c4a2ced4-2c9f-4ae1-adaa-6171833e64df",
    "size": 1,
    "share_type": "25747776-08e5-494f-ab40-a64b9d20d8f7",
    "share_type_name": "default",
    "export_location": "10.254.0.5:/shares/share-50ad5e7b-f6f1-4b78-a651-0812cef2bb67",
    "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "metadata": {},
    "status": "available",
    "access_rules_status": "active",
    "description": "Changed description.",
    "host": "manila2@generic1#GENERIC1",
    "task_state": null,
    "is_public": true,
    "snapshot_support": true,
    "name": "my_share4",
    "has_replicas": false,
    "replication_type": null,
    "created_at": "2015-09-16T17:26:28.000000",
    "share_proto": "NFS",
    "volume_type": "default",
    "source_cgsnapshot_member_id": null
  }
]

```



```
}  
    ]  
}
```

18.3.4. Manage share

Method	URI	Description
POST	/v2/{tenant_id}/shares/manage	Configures Shared File Systems to manage a share. This API is available for micro-versions later than or equal to 2.7

Normal response codes: 200

18.3.4.1. Request

This table shows the header parameters for the manage share request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the manage share request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 18.7. Manage share: JSON request

```
{
  "share": {
    "protocol": "nfs",
    "name": "share_texas1",
    "share_type": "d",
    "is_public": false,
    "driver_options": {
      "opt1": "opt1",
      "opt2": "opt2"
    },
    "export_path": "10.254.0.5:/shares/share-42033c24-0261-424f-abda-4fef2f6dbfd5",
    "service_host": "manila2@unmanage1#UNMANAGE1",
    "description": "Lets manage share."
  }
}
```

18.3.4.2. Response

Example 18.8. Manage share: JSON response

```
{
  "share": {
    "links": [
      {
        "href": "http://172.18.198.54:8786/v2/16e1ab15c35a457e9c2b2aa189f544e1/shares/00137b40-ca06-4ae8-83a3-2c5989eebcce",
        "rel": "self"
      },
      {

```

```

        "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/00137b40-ca06-4ae8-83a3-2c5989eebcce",
        "rel": "bookmark"
    }
],
"availability_zone": null,
"share_network_id": null,
"export_locations": [],
"share_server_id": null,
"snapshot_id": null,
"id": "00137b40-ca06-4ae8-83a3-2c5989eebcce",
"size": null,
"share_type": "14747856-08e5-494f-ab40-a64b9d20d8f7",
"share_type_name": "d",
"export_location": "10.254.0.5:/shares/share-42033c24-0261-424f-
abda-4fef2f6dbfd5",
"consistency_group_id": null,
"project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
"metadata": {},
"status": "manage_starting",
"description": "Lets manage share.",
"host": "manila2@unmanage1#UNMANAGE1",
"access_rules_status": "active",
"has_replicas": false,
"replication_type": null,
"is_public": false,
"snapshot_support": true,
"name": "share_texas1",
"created_at": "2015-09-17T16:21:12.000000",
"share_proto": "NFS",
"volume_type": "d",
"source_cgsnapshot_member_id": null
}
}

```

18.3.5. Show share details

Method	URI	Description
GET	/v2/{tenant_id}/shares/{share_id}	Shows details for a share.

Normal response codes: 200

18.3.5.1. Request

This table shows the header parameters for the show share details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show share details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

This operation does not accept a request body.

18.3.5.2. Response

Example 18.9. Show share details: JSON response

```
{
  "share": {
    "links": [
      {
        "href": "http://172.18.198.54:8786/v2/16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
        "rel": "self"
      },
      {
        "href": "http://172.18.198.54:8786/16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
        "rel": "bookmark"
      }
    ],
    "availability_zone": "nova",
    "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
    "export_locations": [],
    "share_server_id": "e268f4aa-d571-43dd-9ab3-f49ad06ffaef",
    "snapshot_id": null,
    "id": "011d21e2-fbc3-4e4a-9993-9ea223f73264",
    "size": 1,
    "share_type": "25747776-08e5-494f-ab40-a64b9d20d8f7",
    "share_type_name": "default",
    "export_location": null,
    "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
  }
}
```

```
    "metadata": {
      "project": "my_app",
      "aim": "doc"
    },
    "status": "available",
    "description": "My custom share London",
    "host": "manila2@generic1#GENERIC1",
    "access_rules_status": "active",
    "has_replicas": false,
    "replication_type": null,
    "task_state": null,
    "is_public": true,
    "snapshot_support": true,
    "name": "share_London",
    "created_at": "2015-09-18T10:25:24.000000",
    "share_proto": "NFS",
    "volume_type": "default",
    "source_cgsnapshot_member_id": null
  }
}
```

18.3.6. Update share

Method	URI	Description
PUT	/v2/{tenant_id}/shares/{share_id}	Updates a share.

You can update these attributes:

- `display_name`, which also changes the name of the share.
- `display_description`, which also changes the description of the share.
- `is_public`. Changes the level of visibility.

If you try to update other attributes, they retain their previous values.

Normal response codes: 200

18.3.6.1. Request

This table shows the header parameters for the update share request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the update share request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

Example 18.10. Update share: JSON request

```
{
  "share": {
    "is_public": true,
    "display_description": "Changing the share description."
  }
}
```

18.3.6.2. Response

Example 18.11. Update share: JSON response

```
{
  "share": {
    "links": [
      {
        "href": "http://172.18.198.54:8786/v2/16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
        "rel": "self"
      },
    ],
  }
}
```

```
        {
          "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
          "rel": "bookmark"
        }
      ],
      "availability_zone": "nova",
      "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
      "export_locations": [],
      "share_server_id": "e268f4aa-d571-43dd-9ab3-f49ad06ffaef",
      "snapshot_id": null,
      "id": "011d21e2-fbc3-4e4a-9993-9ea223f73264",
      "size": 1,
      "share_type": "25747776-08e5-494f-ab40-a64b9d20d8f7",
      "share_type_name": "default",
      "export_location": null,
      "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
      "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
      "metadata": {
        "project": "my_app",
        "aim": "doc"
      },
      "status": "error",
      "description": "Changing the share description.",
      "host": "manila2@generic1#GENERIC1",
      "task_state": null,
      "is_public": true,
      "snapshot_support": true,
      "name": "share_London",
      "created_at": "2015-09-18T10:25:24.000000",
      "share_proto": "NFS",
      "volume_type": "default",
      "source_cgsnapshot_member_id": null
    }
  }
}
```

18.3.7. Delete share

Method	URI	Description
DELETE	/v2/{tenant_id}/shares/{share_id} {?consistency_group_id}	Deletes a share.

Normal response codes: 202

18.3.7.1. Request

This table shows the header parameters for the delete share request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the delete share request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

This table shows the query parameters for the delete share request:

Name	Type	Description
consistency_group_id	UUID (Optional)	(Since API v2.4) The UUID of the consistency group where the share was created. You can omit this parameter if the share was created without a consistency group.

This operation does not accept a request body.

18.4. Share export locations (since API v2.9)

Set of APIs used for viewing export locations of shares.

By default, these APIs are admin-only. Use the `policy.json` file to grant permissions for these actions to other roles.

Method	URI	Description
GET	/v2/{tenant_id}/shares/{share_id}/export_locations	Lists all export locations for a share.
GET	/v2/{tenant_id}/shares/{share_id}/export_locations/{export_location_id}	Show details of an export location belonging to a share.

18.4.1. List export locations

Method	URI	Description
GET	/v2/{tenant_id}/shares/{share_id}/export_locations	Lists all export locations for a share.

Normal response codes: 200

18.4.1.1. Request

This table shows the header parameters for the list export locations request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list export locations request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

This operation does not accept a request body.

18.4.1.2. Response

Example 18.12. List export locations: JSON response

```
{
  "export_locations": [
    {
      "path": "10.254.0.3:/shares/share-e1c2d35e-fe67-4028-ad7a-45f668732b1d",
      "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
      "is_admin_only": false,
      "id": "b6bd76ce-12a2-42a9-a30a-8a43b503867d",
      "preferred": false
    },
    {
      "path": "10.0.0.3:/shares/share-e1c2d35e-fe67-4028-ad7a-45f668732b1d",
      "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
      "is_admin_only": true,
      "id": "6921e862-88bc-49a5-a2df-efeed9acd583",
      "preferred": false
    }
  ]
}
```

18.4.2. Show single export location

Method	URI	Description
GET	/v2/{tenant_id}/shares/{share_id}/export_locations/{export_location_id}	Show details of an export location belonging to a share.

Normal response codes: 200

18.4.2.1. Request

This table shows the header parameters for the show single export location request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show single export location request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.
{export_location_id}	UUID	The UUID of the export location.

This operation does not accept a request body.

18.4.2.2. Response

Example 18.13. Show single export location: JSON response

```
{
  "export_location": {
    "created_at": "2016-03-24T14:20:47.000000",
    "updated_at": "2016-03-24T14:20:47.000000",
    "preferred": false,
    "is_admin_only": true,
    "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
    "path": "10.0.0.3:/shares/share-e1c2d35e-fe67-4028-ad7a-45f668732b1d",
    "id": "6921e862-88bc-49a5-a2df-efeed9acd583"
  }
}
```

18.5. Share metadata

Shows, sets, updates, and unsets share metadata.

Method	URI	Description
GET	/v2/{tenant_id}/shares/{share_id}/metadata	Shows the metadata for a share.
PUT	/v2/{tenant_id}/shares/{share_id}/metadata	Updates the metadata for a share.

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/metadata	Sets the metadata on a share.
DELETE	/v2/{tenant_id}/shares/{share_id}/metadata/{key}	Unsets the metadata on a share.

18.5.1. Show share metadata

Method	URI	Description
GET	/v2/{tenant_id}/shares/{share_id}/metadata	Shows the metadata for a share.

Normal response codes: 200

18.5.1.1. Request

This table shows the header parameters for the show share metadata request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show share metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

This operation does not accept a request body.

18.5.1.2. Response

Example 18.14. Show share metadata: JSON response

```
{
  "metadata": {
    "project": "my_app",
    "aim": "doc"
  }
}
```

18.5.2. Update share metadata

Method	URI	Description
PUT	/v2/{tenant_id}/shares/{share_id}/metadata	Updates the metadata for a share.

Normal response codes: 200

18.5.2.1. Request

This table shows the header parameters for the update share metadata request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the update share metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

Example 18.15. Update share metadata: JSON request

```
{
  "metadata": {
    "aim": "changed_doc",
    "project": "my_app",
    "new_metadata_key": "new_information"
  }
}
```

18.5.2.2. Response

Example 18.16. Update share metadata: JSON response

```
{
  "metadata": {
    "aim": "changed_doc",
    "project": "my_app",
    "new_metadata_key": "new_information"
  }
}
```

18.5.3. Set share metadata

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/metadata	Sets the metadata on a share.

Normal response codes: 200

18.5.3.1. Request

This table shows the header parameters for the set share metadata request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the set share metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

Example 18.17. Set share metadata: JSON request

```
{
  "metadata": {
    "key1": "value1"
  }
}
```

18.5.3.2. Response

Example 18.18. Set share metadata: JSON response

```
{
  "metadata": {
    "aim": "changed_doc",
    "project": "my_app",
    "key1": "value1",
    "new_metadata_key": "new_information",
    "key": "value"
  }
}
```

18.5.4. Unset share metadata

Method	URI	Description
DELETE	/v2/{tenant_id}/shares/{share_id}/metadata/{key}	Unsets the metadata on a share.

To unset a metadata key value, specify only the key name in the URI.

Normal response codes: 200

18.5.4.1. Request

This table shows the header parameters for the unset share metadata request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the unset share metadata request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.
{metadata_key}	String	The metadata key.

This operation does not accept a request body.

18.6. Share actions

Grants or revokes share access, lists the permissions for a share, and explicitly updates the state of a share.

To grant or revoke share access, specify one of these supported share access levels:

- `rw`. Read and write (RW) access.
- `ro`. Read-only (RO) access.

You must also specify one of these supported authentication methods:

- `ip`. Authenticates an instance through its IP address. A valid format is `XX.XX.XX.XX` or `XX.XX.XX.XX/XX`. For example `0.0.0.0/0`.
- `cert`. Authenticates an instance through a TLS certificate. Specify the TLS identity as the `IDENTKEY`. A valid value is any string up to 64 characters long in the common name (CN) of the certificate. The meaning of a string depends on its interpretation.
- `user`. Authenticates by a user or group name. A valid value is an alphanumeric string that can contain some special characters and is from 4 to 32 characters long.

To verify that the access rules (ACL) were configured correctly for a share, you list permissions for a share.

As administrator, you can reset the state of a share and force-delete a share in any state. Use the `policy.json` file to grant permissions for this action to other roles.

You can set the state of a share to one of these supported states:

- available
- error
- creating
- deleting
- error_deleting



Caution

If micro-version 1.0-2.6 is used then all share actions, defined below, should include prefix `os-` in top element of request JSON's body.

For example: `{"access_list": null}` is valid for v2.7+. And `{"os-access_list": null}` is valid for v1.0-2.6

Method	URI	Description
POST	<code>/v2/{tenant_id}/shares/{share_id}/action</code>	Grants access to a share.
POST	<code>/v2/{tenant_id}/shares/{share_id}/action</code>	Revokes access from a share.
POST	<code>/v2/{tenant_id}/shares/{share_id}/action</code>	Lists access rules for a share.
POST	<code>/v2/{tenant_id}/shares/{share_id}/action</code>	Administrator only. Explicitly updates the state of a share.
POST	<code>/v2/{tenant_id}/shares/{share_id}/action</code>	Administrator only. Force-deletes a share in any state.
POST	<code>/v2/{tenant_id}/shares/{share_id}/action</code>	Increases the size of a share.
POST	<code>/v2/{tenant_id}/shares/{share_id}/action</code>	Shrinks the size of a share.
POST	<code>/v2/{tenant_id}/shares/{share_id}/action</code>	Configures Shared File Systems to stop managing a share.

18.6.1. Grant access

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Grants access to a share.

Normal response codes: 200

18.6.1.1. Request

This table shows the header parameters for the grant access request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the grant access request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

Example 18.19. Grant access: JSON request

```
{
  "allow_access": {
    "access_level": "rw",
    "access_type": "ip",
    "access_to": "0.0.0.0/0"
  }
}
```

18.6.1.2. Response

Example 18.20. Grant access: JSON response

```
{
  "access": {
    "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
    "created_at": "2015-09-07T09:14:48.000000",
    "updated_at": null,
    "access_type": "ip",
    "access_to": "0.0.0.0/0",
    "access_level": "rw",
    "id": "a25b2df3-90bd-4add-afa6-5f0dbbd50452"
  }
}
```

18.6.2. Revoke access

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Revokes access from a share.

Normal response codes: 202

18.6.2.1. Request

This table shows the header parameters for the revoke access request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the revoke access request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

Example 18.21. Revoke access: JSON request

```
{
  "deny_access": {
    "access_id": "a25b2df3-90bd-4add-afa6-5f0dbbd50452"
  }
}
```

18.6.3. List access rules

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Lists access rules for a share.

Normal response codes: 200

18.6.3.1. Request

This table shows the header parameters for the list access rules request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list access rules request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

Example 18.22. List access rules: JSON request

```
{
  "access_list": null
}
```

18.6.3.2. Response

Example 18.23. List access rules: JSON response

```
{
  "access_list": [
    {
      "access_level": "rw",
      "state": "error",
      "id": "507bf114-36f2-4f56-8cf4-857985ca87c1",
      "access_type": "cert",
      "access_to": "example.com"
    },
    {
      "access_level": "rw",
      "state": "active",
      "id": "a25b2df3-90bd-4add-afa6-5f0dbbd50452",
      "access_type": "ip",
      "access_to": "0.0.0.0/0"
    }
  ]
}
```

18.6.4. Reset share state

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Administrator only. Explicitly updates the state of a share.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

18.6.4.1. Request

This table shows the header parameters for the reset share state request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the reset share state request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

Example 18.24. Reset share state: JSON request

```
{
  "reset_status": {
    "status": "error"
  }
}
```

18.6.5. Force-delete share

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Administrator only. Force-deletes a share in any state.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

18.6.5.1. Request

This table shows the header parameters for the force-delete share request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the force-delete share request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

Example 18.25. Force-delete share: JSON request

```
{
  "force_delete": null
}
```

18.6.6. Extend share

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Increases the size of a share.

Normal response codes: 202

18.6.6.1. Request

This table shows the header parameters for the extend share request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the extend share request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

Example 18.26. Extend share: JSON request

```
{
  "extend": {
    "new_size": 2
  }
}
```

18.6.7. Shrink share

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Shrinks the size of a share.

Normal response codes: 202

18.6.7.1. Request

This table shows the header parameters for the shrink share request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the shrink share request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

Example 18.27. Shrink share: JSON request

```
{
  "shrink": {
    "new_size": 1
  }
}
```

18.6.8. Unmanage share

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Configures Shared File Systems to stop managing a share.

Normal response codes: 202

18.6.8.1. Request

This table shows the header parameters for the unmanage share request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the unmanage share request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

Example 18.28. Unmanage share: JSON request

```
{
  "unmanage": null
}
```

18.7. Share snapshots

Use the shared file service to make snapshots of shares. A share snapshot is a point-in-time, read-only copy of the data that is contained in a share. You can create, manage, update, and delete share snapshots. After you create or manage a share snapshot, you can create a share from it.

You can update a share snapshot to rename it, change its description, or update its state to one of these supported states:

- available
- error
- creating
- deleting
- error_deleting
- manage_starting
- manage_error

- `unmanage_starting`
- `unmanage_error`

As administrator, you can also reset the state of a snapshot and force-delete a share snapshot in any state. Use the `policy.json` file to grant permissions for these actions to other roles.

Method	URI	Description
POST	<code>/v2/{tenant_id}/snapshots</code>	Creates a snapshot from a share.
GET	<code>/v2/{tenant_id}/snapshots</code>	Lists all share snapshots.
POST	<code>/v2/{tenant_id}/snapshots/manage</code>	(Since API v2.12) Configures Shared File Systems to manage a share snapshot.
GET	<code>/v2/{tenant_id}/snapshots/detail</code>	Lists all share snapshots with details.
GET	<code>/v2/{tenant_id}/snapshots/{snapshot_id}</code>	Shows details for a share snapshot.
PUT	<code>/v2/{tenant_id}/snapshots/{snapshot_id}</code>	Updates a share snapshot.
DELETE	<code>/v2/{tenant_id}/snapshots/{snapshot_id}</code>	Deletes a share snapshot.
POST	<code>/v2/{tenant_id}/snapshots/{snapshot_id}/action</code>	(Since API v2.12) Configures Shared File Systems to stop managing a share snapshot.
POST	<code>/v2/{tenant_id}/snapshots/{snapshot_id}/action</code>	Administrator only. Explicitly updates the state of a share snapshot.
POST	<code>/v2/{tenant_id}/snapshots/{snapshot_id}/action</code>	Administrator only. Force-deletes a share snapshot in any state.

18.7.1. Create share snapshot

Method	URI	Description
POST	/v2/{tenant_id}/snapshots	Creates a snapshot from a share.

Normal response codes: 200

18.7.1.1. Request

This table shows the header parameters for the create share snapshot request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the create share snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 18.29. Create share snapshot: JSON request

```
{
  "snapshot": {
    "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
    "force": "True",
    "name": "snapshot_share1",
    "description": "Here is a snapshot of share Share1"
  }
}
```

18.7.1.2. Response

Example 18.30. Create share snapshot: JSON response

```
{
  "snapshot": {
    "status": "creating",
    "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
    "name": "snapshot_share1",
    "links": [
      {
        "href": "http://172.18.198.54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/6d221c1d-0200-461e-8d20-24b4776b9ddb",
        "rel": "self"
      },
      {
        "href": "http://172.18.198.54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/6d221c1d-0200-461e-8d20-24b4776b9ddb",
        "rel": "bookmark"
      }
    ]
  }
}
```

```
    ],  
    "created_at": "2015-09-07T11:50:39.756808",  
    "description": "Here is a snapshot of share Share1",  
    "share_proto": "NFS",  
    "share_size": 1,  
    "id": "6d221c1d-0200-461e-8d20-24b4776b9ddb",  
    "size": 1  
  }  
}
```

18.7.2. List share snapshots

Method	URI	Description
GET	/v2/{tenant_id}/snapshots	Lists all share snapshots.

Normal response codes: 200

18.7.2.1. Request

This table shows the header parameters for the list share snapshots request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list share snapshots request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.7.2.2. Response

Example 18.31. List share snapshots: JSON response

```
{
  "snapshots": [
    {
      "id": "086a1aa6-c425-4ecd-9612-391a3b1b9375",
      "links": [
        {
          "href": "http://172.18.198.54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/086a1aa6-c425-4ecd-9612-391a3b1b9375",
          "rel": "self"
        },
        {
          "href": "http://172.18.198.54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/086a1aa6-c425-4ecd-9612-391a3b1b9375",
          "rel": "bookmark"
        }
      ],
      "name": "snapshot_My_share"
    },
    {
      "id": "6d221c1d-0200-461e-8d20-24b4776b9ddb",
      "links": [
        {
          "href": "http://172.18.198.54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/6d221c1d-0200-461e-8d20-24b4776b9ddb",
          "rel": "self"
        }
      ]
    }
  ]
}
```

```
        },  
        {  
            "href": "http://172.18.198.  
54:8786/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/  
6d221c1d-0200-461e-8d20-24b4776b9ddb",  
            "rel": "bookmark"  
        }  
    ],  
    "name": "snapshot_share1"  
}  
]  
}
```

18.7.3. Manage share snapshot

Method	URI	Description
POST	/v2/{tenant_id}/snapshots/manage	(Since API v2.12) Configures Shared File Systems to manage a share snapshot.

Normal response codes: 200

18.7.3.1. Request

This table shows the header parameters for the manage share snapshot request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the manage share snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 18.32. Manage share snapshot: JSON request

```
{
  "snapshot": {
    "share_id": "dd6c5d35-9db1-4662-a7ae-8b52f880aeba",
    "provider_location": "4045fee5-4e0e-408e-97f3-15e25239dbc9",
    "name": "managed_snapshot",
    "description": "description_of_managed_snapshot",
    "driver_options": {
      "opt1": "opt1",
      "opt2": "opt2"
    }
  }
}
```

18.7.3.2. Response

Example 18.33. Manage share snapshot: JSON response

```
{
  "snapshot": {
    "id": "22de7000-3a32-4fe1-bd0c-38d03f93dec3",
    "share_id": "dd6c5d35-9db1-4662-a7ae-8b52f880aeba",
    "share_size": 1,
    "created_at": "2016-04-01T15:16:17.000000",
    "status": "manage_starting",
    "name": "managed_snapshot",
    "description": "description_of_managed_snapshot",
    "size": 1,
    "share_proto": "NFS",
    "links": [
      {

```

```
        "href": "http://127.0.0.1:8786/v2/
907004508ef4447397ce6741a8f037c1/snapshots/22de7000-3a32-4fe1-
bd0c-38d03f93dec3",
        "rel": "self"
    },
    {
        "href": "http://127.0.0.1:8786/
907004508ef4447397ce6741a8f037c1/snapshots/22de7000-3a32-4fe1-
bd0c-38d03f93dec3",
        "rel": "bookmark"
    }
],
"provider_location": "4045fee5-4e0e-408e-97f3-15e25239dbc9"
}
```

18.7.4. List share snapshots with details

Method	URI	Description
GET	/v2/{tenant_id}/snapshots/detail	Lists all share snapshots with details.

Normal response codes: 200

18.7.4.1. Request

This table shows the header parameters for the list share snapshots with details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list share snapshots with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.7.4.2. Response

Example 18.34. List share snapshots with details: JSON response

```
{
  "snapshots": [
    {
      "status": "creating",
      "share_id": "d94a8548-2079-4be0-b21c-0a887acd31ca",
      "name": "snapshot_My_share",
      "links": [
        {
          "href": "http://172.18.198.54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/086a1aa6-c425-4ecd-9612-391a3b1b9375",
          "rel": "self"
        },
        {
          "href": "http://172.18.198.54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/086a1aa6-c425-4ecd-9612-391a3b1b9375",
          "rel": "bookmark"
        }
      ],
      "created_at": "2015-09-07T11:55:09.000000",
      "description": "Here is a snapshot of share My_share",
      "share_proto": "NFS",
      "share_size": 1,
      "id": "086a1aa6-c425-4ecd-9612-391a3b1b9375",
      "size": 1
    },
    {

```



```
    "status": "available",
    "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
    "name": "snapshot_share1",
    "links": [
      {
        "href": "http://172.18.198.
54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
        "rel": "self"
      },
      {
        "href": "http://172.18.198.
54:8786/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
        "rel": "bookmark"
      }
    ],
    "created_at": "2015-09-07T11:50:39.000000",
    "description": "Here is a snapshot of share Share1",
    "share_proto": "NFS",
    "share_size": 1,
    "id": "6d221c1d-0200-461e-8d20-24b4776b9ddb",
    "size": 1
  }
]
```

18.7.5. Show share snapshot details

Method	URI	Description
GET	/v2/{tenant_id}/snapshots/{snapshot_id}	Shows details for a share snapshot.

Normal response codes: 200

18.7.5.1. Request

This table shows the header parameters for the show share snapshot details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show share snapshot details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

18.7.5.2. Response

Example 18.35. Show share snapshot details: JSON response

```
{
  "snapshot": {
    "status": "available",
    "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
    "name": "snapshot_share1",
    "links": [
      {
        "href": "http://172.18.198.54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/6d221c1d-0200-461e-8d20-24b4776b9ddb",
        "rel": "self"
      },
      {
        "href": "http://172.18.198.54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/6d221c1d-0200-461e-8d20-24b4776b9ddb",
        "rel": "bookmark"
      }
    ],
    "created_at": "2015-09-07T11:50:39.000000",
    "description": "Here is a snapshot of share Share1",
    "share_proto": "NFS",
    "share_size": 1,
    "id": "6d221c1d-0200-461e-8d20-24b4776b9ddb",
    "size": 1
  }
}
```

```
}  
}
```

18.7.6. Update share snapshot

Method	URI	Description
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}	Updates a share snapshot.

You can update these attributes:

- `display_name`, which also changes the name of the share snapshot.
- `display_description`, which also changes the description of the share snapshot.
- `is_public`. Changes the level of visibility.

If you try to update other attributes, they retain their previous values.

Normal response codes: 200

18.7.6.1. Request

This table shows the header parameters for the update share snapshot request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the update share snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

Example 18.36. Update share snapshot: JSON request

```
{
  "snapshot": {
    "display_name": "snapshot_Share1",
    "display_description": "I am changing a description also. Here is a
snapshot of share Share1"
  }
}
```

18.7.6.2. Response

Example 18.37. Update share snapshot: JSON response

```
{
  "snapshot": {
    "status": "available",
    "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
    "name": "snapshot_Share1",
    "links": [
```

```
        {
          "href": "http://172.18.198.54:8786/
v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
          "rel": "self"
        },
        {
          "href": "http://172.18.198.
54:8786/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
          "rel": "bookmark"
        }
      ],
      "created_at": "2015-09-07T11:50:39.000000",
      "description": "I am changing a description also. Here is a snapshot
of share Share1",
      "share_proto": "NFS",
      "share_size": 1,
      "id": "6d221c1d-0200-461e-8d20-24b4776b9ddb",
      "size": 1
    }
  }
```

18.7.7. Delete share snapshot

Method	URI	Description
DELETE	/v2/{tenant_id}/snapshots/{snapshot_id}	Deletes a share snapshot.

Normal response codes: 202

18.7.7.1. Request

This table shows the header parameters for the delete share snapshot request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the delete share snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

18.7.8. Unmanage share snapshot

Method	URI	Description
POST	/v2/{tenant_id}/snapshots/{snapshot_id}/action	(Since API v2.12) Configures Shared File Systems to stop managing a share snapshot.

Normal response codes: 202

18.7.8.1. Request

This table shows the header parameters for the unmanage share snapshot request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the unmanage share snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

Example 18.38. Unmanage share snapshot: JSON request

```
{
  "unmanage": null
}
```

18.7.9. Reset share snapshot state

Method	URI	Description
POST	/v2/{tenant_id}/snapshots/{snapshot_id}/action	Administrator only. Explicitly updates the state of a share snapshot.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

18.7.9.1. Request

This table shows the header parameters for the reset share snapshot state request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the reset share snapshot state request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

Example 18.39. Reset share snapshot state: JSON request

```
{
  "reset_status": {
    "status": "error"
  }
}
```


18.7.10. Force-delete share snapshot

Method	URI	Description
POST	/v2/{tenant_id}/snapshots/{snapshot_id}/action	Administrator only. Force-deletes a share snapshot in any state.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

18.7.10.1. Request

This table shows the header parameters for the force-delete share snapshot request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the force-delete share snapshot request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{snapshot_id}	UUID	The UUID of the snapshot.

Example 18.40. Force-delete share snapshot: JSON request

```
{
  "force_delete": null
}
```

18.8. Share networks

A share network stores network information that share servers can use where shares are hosted. You can associate a share with a single share network. When you create a share, you can optionally specify the ID of a share network through which instances can access the share.

You can create, update, view, and delete a share network.

When you create a share network, you can specify only one type of network:

- Neutron network. Specify a network ID and subnet ID.
- Nova network. Specify a network ID.

For more information about supported plug-ins for share networks, see [Manila Network Plugins](#).

A share network has these attributes:

- The IP block in Classless Inter-Domain Routing (CIDR) notation from which to allocate the network.

- The IP version of the network.
- The network type, which is `vlan`, `vxlan`, `gre`, or `flat`.
- If the network uses segmentation, a segmentation identifier. For example, VLAN, VXLAN, and GRE networks use segmentation.

Method	URI	Description
POST	/v2/{tenant_id}/share-networks	Creates a share network.
GET	/v2/{tenant_id}/share-networks	Lists all share networks.
GET	/v2/{tenant_id}/share-networks/detail	Lists all share networks with details.
GET	/v2/{tenant_id}/share-networks/{share_network_id}	Shows details for a share network.
PUT	/v2/{tenant_id}/share-networks/{share_network_id}	Updates a share network.
DELETE	/v2/{tenant_id}/share-networks/{share_network_id}	Deletes a share network.
POST	/v2/{tenant_id}/share-networks/{share_network_id}/action	Adds a security service to a share network.
POST	/v2/{tenant_id}/share-networks/{share_network_id}/action	Removes a security service from a share network.

18.8.1. Create share network

Method	URI	Description
POST	/v2/{tenant_id}/share-networks	Creates a share network.

Normal response codes: 200

18.8.1.1. Request

This table shows the header parameters for the create share network request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the create share network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 18.41. Create share network: JSON request

```
{
  "share_network": {
    "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
    "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
    "name": "my_network",
    "description": "This is my share network"
  }
}
```

18.8.1.2. Response

Example 18.42. Create share network: JSON response

```
{
  "share_network": {
    "name": "my_network",
    "segmentation_id": null,
    "created_at": "2015-09-07T14:37:00.583656",
    "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
    "updated_at": null,
    "id": "77eb3421-4549-4789-ac39-0d5185d68c29",
    "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
    "ip_version": null,
    "nova_net_id": null,
    "cidr": null,
    "project_id": "e10a683c20da41248cfd5e1ab3d88c62",
    "network_type": null,
    "description": "This is my share network"
  }
}
```

18.8.2. List share networks

Method	URI	Description
GET	/v2/{tenant_id}/share-networks	Lists all share networks.

Normal response codes: 200

18.8.2.1. Request

This table shows the header parameters for the list share networks request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list share networks request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.8.2.2. Response

Example 18.43. List share networks: JSON response

```
{
  "share_networks": [
    {
      "id": "32763294-e3d4-456a-998d-60047677c2fb",
      "name": "net_my1"
    },
    {
      "id": "713df749-aac0-4a54-af52-10f6c991e80c",
      "name": "net_my"
    },
    {
      "id": "fa158a3d-6d9f-4187-9ca5-abbb82646eb2",
      "name": null
    }
  ]
}
```

18.8.3. List share networks with details

Method	URI	Description
GET	/v2/{tenant_id}/share-networks/detail	Lists all share networks with details.

Normal response codes: 200

18.8.3.1. Request

This table shows the header parameters for the list share networks with details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list share networks with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.8.3.2. Response

Example 18.44. List share networks with details: JSON response

```
{
  "share_networks": [
    {
      "name": "net_my1",
      "segmentation_id": null,
      "created_at": "2015-09-04T14:57:13.000000",
      "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
      "updated_at": null,
      "id": "32763294-e3d4-456a-998d-60047677c2fb",
      "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
      "ip_version": null,
      "nova_net_id": null,
      "cidr": null,
      "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
      "network_type": null,
      "description": "descr"
    },
    {
      "name": "net_my",
      "segmentation_id": null,
      "created_at": "2015-09-04T14:54:25.000000",
      "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
      "updated_at": null,
      "id": "713df749-aac0-4a54-af52-10f6c991e80c",
      "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
      "ip_version": null,
      "nova_net_id": null,

```

```
        "cidr": null,  
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",  
        "network_type": null,  
        "description": "desecr"  
    },  
    {  
        "name": null,  
        "segmentation_id": null,  
        "created_at": "2015-09-04T14:51:41.000000",  
        "neutron_subnet_id": null,  
        "updated_at": null,  
        "id": "fa158a3d-6d9f-4187-9ca5-abbb82646eb2",  
        "neutron_net_id": null,  
        "ip_version": null,  
        "nova_net_id": null,  
        "cidr": null,  
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",  
        "network_type": null,  
        "description": null  
    }  
]  
}
```

18.8.4. Show share network details

Method	URI	Description
GET	/v2/{tenant_id}/share-networks/{share_network_id}	Shows details for a share network.

Normal response codes: 200

18.8.4.1. Request

This table shows the header parameters for the show share network details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show share network details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_network_id}	UUID	The UUID of the share network.

This operation does not accept a request body.

18.8.4.2. Response

Example 18.45. Show share network details: JSON response

```
{
  "share_network": {
    "name": "net_my1",
    "segmentation_id": null,
    "created_at": "2015-09-04T14:56:45.000000",
    "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
    "updated_at": null,
    "id": "7f950b52-6141-4a08-bbb5-bb7ffa3ea5fd",
    "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
    "ip_version": null,
    "nova_net_id": null,
    "cidr": null,
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "network_type": null,
    "description": "descr"
  }
}
```

18.8.5. Update share network

Method	URI	Description
PUT	/v2/{tenant_id}/share-networks/{share_network_id}	Updates a share network.

Note that if the share network is used by any share server, you can update only the name and description attributes.

Normal response codes: 200

18.8.5.1. Request

This table shows the header parameters for the update share network request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the update share network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_network_id}	UUID	The UUID of the share network.

Example 18.46. Update share network: JSON request

```
{
  "share_network": {
    "description": "i'm adding a description"
  }
}
```

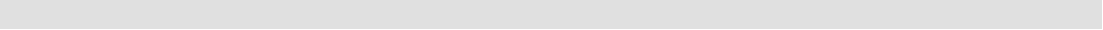
18.8.5.2. Response

Example 18.47. Update share network: JSON response

```
{
  "share_network": {
    "name": "net_my",
    "segmentation_id": null,
    "created_at": "2015-09-04T14:54:25.000000",
    "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
    "updated_at": "2015-09-07T08:02:53.512184",
    "id": "713df749-aac0-4a54-af52-10f6c991e80c",
    "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
    "ip_version": "4",
    "nova_net_id": null,
    "cidr": null,
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "network_type": null,
    "description": "i'm adding a description"
  }
}
```



```
}
```



18.8.6. Delete share network

Method	URI	Description
DELETE	/v2/{tenant_id}/share-networks/{share_network_id}	Deletes a share network.

Normal response codes: 202

18.8.6.1. Request

This table shows the header parameters for the delete share network request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the delete share network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_network_id}	UUID	The UUID of the share network.

This operation does not accept a request body.

18.8.7. Add security service to share network

Method	URI	Description
POST	/v2/{tenant_id}/share-net-works/{share_network_id}/action	Adds a security service to a share network.

Normal response codes: 200

18.8.7.1. Request

This table shows the header parameters for the add security service to share network request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the add security service to share network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_network_id}	UUID	The UUID of the share network.

Example 18.48. Add security service to share network: JSON request

```
{
  "add_security_service": {
    "security_service_id": "3c829734-0679-4c17-9637-801da48c0d5f"
  }
}
```

18.8.7.2. Response

Example 18.49. Add security service to share network: JSON response

```
{
  "share_network": {
    "name": "net2",
    "segmentation_id": null,
    "created_at": "2015-09-07T12:31:12.000000",
    "neutron_subnet_id": null,
    "updated_at": null,
    "id": "d8ae6799-2567-4a89-aafb-fa4424350d2b",
    "neutron_net_id": null,
    "ip_version": null,
    "nova_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
    "cidr": null,
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "network_type": null,
    "description": null
  }
}
```

18.8.8. Remove security service from share network

Method	URI	Description
POST	/v2/{tenant_id}/share-networks/{share_network_id}/action	Removes a security service from a share network.

Normal response codes: 200

18.8.8.1. Request

This table shows the header parameters for the remove security service from share network request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the remove security service from share network request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_network_id}	UUID	The UUID of the share network.

Example 18.50. Remove security service from share network: JSON request

```
{
  "remove_security_service": {
    "security_service_id": "3c829734-0679-4c17-9637-801da48c0d5f"
  }
}
```

18.8.8.2. Response

Example 18.51. Remove security service from share network: JSON response

```
{
  "share_network": {
    "name": "net2",
    "segmentation_id": null,
    "created_at": "2015-09-07T12:31:12.000000",
    "neutron_subnet_id": null,
    "updated_at": null,
    "id": "d8ae6799-2567-4a89-aafb-fa4424350d2b",
    "neutron_net_id": null,
    "ip_version": null,
    "nova_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
    "cidr": null,
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "network_type": null,
    "description": null
  }
}
```

18.9. Security services

You can create, update, view, and delete a security service. A security service stores configuration information for clients for authentication and authorization (AuthN/AuthZ). For example, a share server will be the client for an existing service such as LDAP, Kerberos, or Microsoft Active Directory.

You can associate a share with from one to three security service types:

- `ldap`. LDAP.
- `kerberos`. Kerberos.
- `active_directory`. Microsoft Active Directory.

You can configure a security service with these options:

- A DNS IP address.
- An IP address or host name.
- A domain.
- A user or group name.
- The password for the user, if you specify a user name.

Method	URI	Description
POST	<code>/v2/{tenant_id}/security-services</code>	Creates a security service.
GET	<code>/v2/{tenant_id}/security-services</code>	Lists all security services.
GET	<code>/v2/{tenant_id}/security-services/detail</code>	Lists all security services with details.
GET	<code>/v2/{tenant_id}/security-services/{security_service_id}</code>	Shows details for a security service.
PUT	<code>/v2/{tenant_id}/security-services/{security_service_id}</code>	Updates a security service.
DELETE	<code>/v2/{tenant_id}/security-services/{security_service_id}</code>	Deletes a security service.

18.9.1. Create security service

Method	URI	Description
POST	/v2/{tenant_id}/security-services	Creates a security service.

Normal response codes: 200

18.9.1.1. Request

This table shows the header parameters for the create security service request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the create security service request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 18.52. Create security service: JSON request

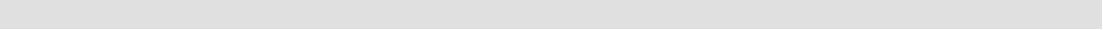
```
{
  "security_service": {
    "description": "Creating my first Security Service",
    "dns_ip": "10.0.0.0/24",
    "user": "demo",
    "password": "****",
    "type": "kerberos",
    "name": "SecServ1"
  }
}
```

18.9.1.2. Response

Example 18.53. Create security service: JSON response

```
{
  "security_service": {
    "status": "new",
    "domain": null,
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "name": "SecServ1",
    "created_at": "2015-09-07T12:19:10.695211",
    "updated_at": null,
    "server": null,
    "dns_ip": "10.0.0.0/24",
    "user": "demo",
    "password": "supersecret",
    "type": "kerberos",
    "id": "3c829734-0679-4c17-9637-801da48c0d5f",
    "description": "Creating my first Security Service"
  }
}
```

```
}
```



18.9.2. List security services

Method	URI	Description
GET	/v2/{tenant_id}/security-services	Lists all security services.

Normal response codes: 200

18.9.2.1. Request

This table shows the header parameters for the list security services request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list security services request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.9.2.2. Response

Example 18.54. List security services: JSON response

```
{
  "security_services": [
    {
      "status": "new",
      "type": "kerberos",
      "id": "3c829734-0679-4c17-9637-801da48c0d5f",
      "name": "SecServ1"
    },
    {
      "status": "new",
      "type": "ldap",
      "id": "5a1d3a12-34a7-4087-8983-50e9ed03509a",
      "name": "SecServ2"
    }
  ]
}
```


18.9.3. List security services with details

Method	URI	Description
GET	/v2/{tenant_id}/security-services/detail	Lists all security services with details.

Normal response codes: 200

18.9.3.1. Request

This table shows the header parameters for the list security services with details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list security services with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.9.3.2. Response

Example 18.55. List security services with details: JSON response

```
{
  "security_services": [
    {
      "status": "new",
      "domain": null,
      "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
      "name": "SecServ1",
      "created_at": "2015-09-07T12:19:10.000000",
      "description": "Creating my first Security Service",
      "updated_at": null,
      "server": null,
      "dns_ip": "10.0.0.0/24",
      "user": "demo",
      "password": "supersecret",
      "type": "kerberos",
      "id": "3c829734-0679-4c17-9637-801da48c0d5f",
      "share_networks": []
    },
    {
      "status": "new",
      "domain": null,
      "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
      "name": "SecServ2",
      "created_at": "2015-09-07T12:25:03.000000",
      "description": "Creating my second Security Service",
      "updated_at": null,
      "server": null,

```

```
        "dns_ip": "10.0.0.0/24",  
        "user": null,  
        "password": null,  
        "type": "ldap",  
        "id": "5a1d3a12-34a7-4087-8983-50e9ed03509a",  
        "share_networks": []  
    }  
]  
}
```

18.9.4. Show security service details

Method	URI	Description
GET	/v2/{tenant_id}/security-services/{security_service_id}	Shows details for a security service.

Normal response codes: 200

18.9.4.1. Request

This table shows the header parameters for the show security service details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show security service details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{security_service_id}	UUID	The UUID of the security service.

This operation does not accept a request body.

18.9.4.2. Response

Example 18.56. Show security service details: JSON response

```
{
  "security_service": {
    "status": "new",
    "domain": null,
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "name": "SecServ1",
    "created_at": "2015-09-07T12:19:10.000000",
    "updated_at": null,
    "server": null,
    "dns_ip": "10.0.0.0/24",
    "user": "demo",
    "password": "supersecret",
    "type": "kerberos",
    "id": "3c829734-0679-4c17-9637-801da48c0d5f",
    "description": "Creating my first Security Service"
  }
}
```

18.9.5. Update security service

Method	URI	Description
PUT	/v2/{tenant_id}/security-services/{security_service_id}	Updates a security service.

If the security service is in active state, you can update only the name and description attributes. A security service in active state is attached to a share network with an associated share server.

Normal response codes: 200

18.9.5.1. Request

This table shows the header parameters for the update security service request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the update security service request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{security_service_id}	UUID	The UUID of the security service.

Example 18.57. Update security service: JSON request

```
{
  "security_service": {
    "domain": "my_domain",
    "password": "****",
    "user": "new_user",
    "description": "Adding a description"
  }
}
```

18.9.5.2. Response

Example 18.58. Update security service: JSON response

```
{
  "security_service": {
    "status": "new",
    "domain": "my_domain",
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "name": "SecServ1",
    "created_at": "2015-09-07T12:19:10.000000",
    "updated_at": "2015-09-07T12:47:21.858737",
    "server": null,
    "dns_ip": "10.0.0.0/24",
    "user": "new_user",
    "password": "pass",
  }
}
```

```
    "type": "kerberos",  
    "id": "3c829734-0679-4c17-9637-801da48c0d5f",  
    "description": "Adding a description"  
  }  
}
```

18.9.6. Delete security service

Method	URI	Description
DELETE	/v2/{tenant_id}/security-services/{security_service_id}	Deletes a security service.

Normal response codes: 202

18.9.6.1. Request

This table shows the header parameters for the delete security service request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the delete security service request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{security_service_id}	UUID	The UUID of the security service.

This operation does not accept a request body.

18.10. Share servers

A share server is created by multi-tenant back-end drivers where shares are hosted. For example, with the generic driver, shares are hosted on Compute VMs. With the cluster_mode driver from NetApp, shares are hosted on virtual storage servers, also known as Vservers or SVMs.

Administrators can perform read and delete actions for share servers. An administrator can delete an active share server only if it contains no dependent shares. If an administrator deletes the share server, the Shared File Systems service creates a share server in response to a subsequent create share request.

An administrator can use the policy.json file to grant permissions for share server actions to other roles.

The status of a share server indicates its current state. After you successfully set up a share server, its status is active. If errors occur during set up such as when server data is not valid, its status is error.

The possible share servers statuses are:

Table 18.2. Share server statuses

Status	Description
active	Share server was successfully set up.
error	The set up or deletion of the share server failed.

Status	Description
deleting	The share server has no dependent shares and is being deleted.
creating	The share server is being created on the back end with data from the database.

Method	URI	Description
GET	/v2/{tenant_id}/share-servers	Lists all share servers.
GET	/v2/{tenant_id}/share-servers/{share_server_id}	Shows details for a share server.
DELETE	/v2/{tenant_id}/share-servers/{share_server_id}	Deletes a share server.

18.10.1. List share servers

Method	URI	Description
GET	/v2/{tenant_id}/share-servers	Lists all share servers.

Normal response codes: 200

18.10.1.1. Request

This table shows the header parameters for the list share servers request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list share servers request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.10.1.2. Response

Example 18.59. List share servers: JSON response

```
{
  "share_servers": [
    {
      "status": "active",
      "updated_at": "2015-09-07T08:52:15.000000",
      "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
      "host": "manila2@generic1",
      "share_network_name": "net_my",
      "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
      "id": "ba11930a-bf1a-4aa7-bae4-a8dfbbaa3cc73"
    }
  ]
}
```


18.10.2. Show share server details

Method	URI	Description
GET	/v2/{tenant_id}/share-servers/{share_server_id}	Shows details for a share server.

Normal response codes: 200

18.10.2.1. Request

This table shows the header parameters for the show share server details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show share server details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_server_id}	UUID	The UUID of the share server.

This operation does not accept a request body.

18.10.2.2. Response

Example 18.60. Show share server details: JSON response

```
{
  "share_server": {
    "status": "active",
    "backend_details": {
      "username": "manila",
      "router_id": "4b62ce91-56c5-45c1-b0ef-8cbbe5dd34f4",
      "pk_path": "/opt/stack/.ssh/id_rsa",
      "subnet_id": "16e99ad6-5191-461c-9f34-ac84a39c3adb",
      "ip": "10.254.0.3",
      "instance_id": "75f2f282-af65-49ba-a7b1-525705b1bf1a",
      "public_address": "10.254.0.3",
      "service_port_id": "8ff21760-961e-4b83-a032-03fd559bb1d3"
    },
    "created_at": "2015-09-07T08:37:19.000000",
    "updated_at": "2015-09-07T08:52:15.000000",
    "share_network_name": "net_my",
    "host": "manila2@generic1",
    "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "id": "ba11930a-bf1a-4aa7-bae4-a8dfbaa3cc73"
  }
}
```

18.10.3. Delete share server

Method	URI	Description
DELETE	/v2/{tenant_id}/share-servers/{share_server_id}	Deletes a share server.

An administrator can delete an active share server only if it contains no dependent shares.

Normal response codes: 202

18.10.3.1. Request

This table shows the header parameters for the delete share server request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the delete share server request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_server_id}	UUID	The UUID of the share server.

This operation does not accept a request body.

18.11. Share instances (since API v2.3)

Administrators can list, show information for, explicitly set the state of, and force-delete share instances. Use the `policy.json` file to grant permissions for these actions to other roles.

Method	URI	Description
GET	/v2/{tenant_id}/share_instances	Lists all share instances.
GET	/v2/{tenant_id}/share_instances/{share_instance_id}	Shows details for a share instance.
POST	/v2/{tenant_id}/share_instances/{share_instance_id}/action	Administrator only. Explicitly updates the state of a share instance.
POST	/v2/{tenant_id}/share_instances/{share_instance_id}/action	Administrator only. Force-deletes a share instance.

18.11.1. List share instances

Method	URI	Description
GET	/v2/{tenant_id}/share_instances	Lists all share instances.

Normal response codes: 200

18.11.1.1. Request

This table shows the header parameters for the list share instances request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list share instances request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.11.1.2. Response

Example 18.61. List share instances: JSON response

```
{
  "share_instances": [
    {
      "status": "error",
      "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
      "availability_zone": "nova",
      "replica_state": null,
      "created_at": "2015-09-07T08:41:20.000000",
      "export_location": "10.254.0.3:/shares/share-081f7030-c54f-42f5-98ee-93a37393e0f2",
      "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
      "export_locations": [
        "10.254.0.3:/shares/share-081f7030-c54f-42f5-98ee-93a37393e0f2"
      ],
      "share_server_id": "ba11930a-bf1a-4aa7-bae4-a8dfbaa3cc73",
      "host": "manila2@generic1#GENERIC1",
      "id": "081f7030-c54f-42f5-98ee-93a37393e0f2"
    },
    {
      "status": "available",
      "share_id": "d94a8548-2079-4be0-b21c-0a887acd31ca",
      "availability_zone": "nova",
      "replica_state": null,
      "created_at": "2015-09-07T08:51:34.000000",
      "export_location": "10.254.0.3:/shares/share-75559a8b-c90c-42a7-bda2-edbe86acfb7b",
      "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
    }
  ]
}
```

```
        "export_locations": [  
            "10.254.0.3:/shares/share-75559a8b-c90c-42a7-bda2-  
edbe86acfb7b"  
        ],  
        "share_server_id": "ba11930a-bf1a-4aa7-bae4-a8dfbaa3cc73",  
        "host": "manila2@generic1#GENERIC1",  
        "id": "75559a8b-c90c-42a7-bda2-edbe86acfb7b"  
    }  
]  
}
```

18.11.2. Show share instance details

Method	URI	Description
GET	/v2/{tenant_id}/share_instances/{share_instance_id}	Shows details for a share instance.

Normal response codes: 200

18.11.2.1. Request

This table shows the header parameters for the show share instance details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show share instance details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_instance_id}	UUID	The UUID of the share instance.

This operation does not accept a request body.

18.11.2.2. Response

Example 18.62. Show share instance details: JSON response

```
{
  "share_instance": {
    "status": "available",
    "share_id": "d94a8548-2079-4be0-b21c-0a887acd31ca",
    "availability_zone": "nova",
    "replica_state": null,
    "created_at": "2015-09-07T08:51:34.000000",
    "export_location": "10.254.0.3:/shares/share-75559a8b-c90c-42a7-bda2-edbe86acfb7b",
    "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
    "export_locations": [
      "10.254.0.3:/shares/share-75559a8b-c90c-42a7-bda2-edbe86acfb7b"
    ],
    "share_server_id": "ba11930a-bf1a-4aa7-bae4-a8dfbaa3cc73",
    "host": "manila2@generic1#GENERIC1",
    "access_rules_status": "active",
    "id": "75559a8b-c90c-42a7-bda2-edbe86acfb7b"
  }
}
```

18.11.3. Reset share instance state

Method	URI	Description
POST	/v2/{tenant_id}/share_instances/{share_instance_id}/action	Administrator only. Explicitly updates the state of a share instance.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

18.11.3.1. Request

This table shows the header parameters for the reset share instance state request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the reset share instance state request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_instance_id}	UUID	The UUID of the share instance.

Example 18.63. Reset share instance state: JSON request

```
{
  "reset_status": {
    "status": "available"
  }
}
```

18.11.4. Force-delete share instance

Method	URI	Description
POST	/v2/{tenant_id}/share_instances/{share_instance_id}/action	Administrator only. Force-deletes a share instance.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

18.11.4.1. Request

This table shows the header parameters for the force-delete share instance request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the force-delete share instance request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_instance_id}	UUID	The UUID of the share instance.

Example 18.64. Force-delete share instance: JSON request

```
{
  "force_delete": null
}
```

18.12. Share instance export locations (since API v2.9)

Set of APIs used to view export locations of share instances.

By default, these APIs are admin-only. Use the `policy.json` file to grant permissions for these actions to other roles.

Method	URI	Description
GET	/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations	Lists all export locations for a share instance.
GET	/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations/{export_location_id}	Show details of an export location belonging to a share instance.

18.12.1. List export locations

Method	URI	Description
GET	/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations	Lists all export locations for a share instance.

Normal response codes: 200

18.12.1.1. Request

This table shows the header parameters for the list export locations request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list export locations request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_instance_id}	UUID	The UUID of the share instance.

This operation does not accept a request body.

18.12.1.2. Response

Example 18.65. List export locations: JSON response

```
{
  "export_locations": [
    {
      "path": "10.254.0.3:/shares/share-e1c2d35e-fe67-4028-ad7a-45f668732b1d",
      "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
      "is_admin_only": false,
      "id": "b6bd76ce-12a2-42a9-a30a-8a43b503867d",
      "preferred": false
    },
    {
      "path": "10.0.0.3:/shares/share-e1c2d35e-fe67-4028-ad7a-45f668732b1d",
      "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
      "is_admin_only": true,
      "id": "6921e862-88bc-49a5-a2df-efeed9acd583",
      "preferred": false
    }
  ]
}
```


18.12.2. Show single export location

Method	URI	Description
GET	/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations/{export_location_id}	Show details of an export location belonging to a share instance.

Normal response codes: 200

18.12.2.1. Request

This table shows the header parameters for the show single export location request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show single export location request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_instance_id}	UUID	The UUID of the share instance.
{export_location_id}	UUID	The UUID of the export location.

This operation does not accept a request body.

18.12.2.2. Response

Example 18.66. Show single export location: JSON response

```
{
  "export_location": {
    "created_at": "2016-03-24T14:20:47.000000",
    "updated_at": "2016-03-24T14:20:47.000000",
    "preferred": false,
    "is_admin_only": true,
    "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
    "path": "10.0.0.3:/shares/share-e1c2d35e-fe67-4028-ad7a-45f668732b1d",
    "id": "6921e862-88bc-49a5-a2df-efeed9acd583"
  }
}
```

18.13. Share types

A share type enables you to filter or choose back ends before you create a share. A share type behaves in the same way as a Block Storage volume type behaves.

You set a share type to private or public and manage the access to the private share types.

When you issue a create a share type request, you can submit a request body with either a `share_type` or `volume_type` object. The use of the `volume_type` object is deprecated but supported. It is recommended that you use the `share_type` object when you create a share type.

No matter which object type you include in the request, the API creates both a `volume_type` object and a `share_type` object. Both objects have the same ID. When you issue a list share types request, the response shows both `share_types` and `volume_types` objects.

You can set share types as either public or private. By default a share type is created as publicly accessible. Set `share_type_access:is_public` (`os-share-type-access:is_public` for micro-versions 1.0-2.6) to `False` to make the share type private.

You can manage the access to the private share types for the different projects. You can add access, remove access, and get information about access for a private share type.

Administrators can create share types with these extra specifications that are used to filter back ends:

- `driver_handles_share_servers`. Required. Defines the driver mode for share server, or storage, life cycle management. The Shared File Systems service creates a share server for the export of shares.

Set to `True` when the share driver manages, or handles, the share server life cycle.

Set to `False` when an administrator rather than a share driver manages the storage life cycle.

- `snapshot_support`. Filters back ends by whether they do or do not support share snapshots.

Set to `True` to find back ends that support share snapshots.

Set to `False` to find back ends that do not support share snapshots.

Administrators can also set additional extra specifications for a share type for the following purposes:

- Filter back ends. Specify these unqualified extra specifications in this format: `extra_spec=value`. For example, `netapp_raid_type=raid4`.
- Set data for the driver. Except for the special capabilities prefix, you specify these qualified extra specifications with its prefix followed by a colon: `vendor:extra_spec=value`. For example, `netapp:thin_provisioned=true`.

The scheduler uses the special capabilities prefix for filtering. The scheduler can only create a share on a back end that reports capabilities that match the unscoped extra-spec keys for the share type. For details, see [Capabilities and Extra-Specs](#).

Each driver implementation determines which extra specification keys it uses. For details, see the documentation for the driver.

An administrator can use the `policy.json` file to grant permissions for share type creation with extra specifications to other roles.

Method	URI	Description
POST	/v2/{tenant_id}/types	Creates a share type.
GET	/v2/{tenant_id}/types	Lists all share types.
GET	/v2/{tenant_id}/types/default	Lists default share types.
DELETE	/v2/{tenant_id}/types/{share_type_id}	Deletes a share type.
GET	/v2/{tenant_id}/types/{share_type_id}/extra_specs	Lists the extra specifications for a share type.
POST	/v2/{tenant_id}/types/{share_type_id}/extra_specs	Sets an extra specification for the share type.
DELETE	/v2/{tenant_id}/types/{share_type_id}/extra_specs/{key}	Unsets an extra specification for the share type.
POST	/v2/{tenant_id}/types/{share_type_id}/action	Adds share type access for a project.
POST	/v2/{tenant_id}/types/{share_type_id}/action	Removes share type access from a project.
GET	/v2/{tenant_id}/types/{share_type_id}/share_type_access	Shows access details for a share type.

18.13.1. Create share type

Method	URI	Description
POST	/v2/{tenant_id}/types	Creates a share type.

Normal response codes: 200

18.13.1.1. Request

This table shows the header parameters for the create share type request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the create share type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 18.67. Create share type: JSON request

```
{
  "volume_type": {
    "os-share-type-access:is_public": true,
    "extra_specs": {
      "driver_handles_share_servers": true,
      "snapshot_support": true
    },
    "name": "my_new_volume_type"
  }
}
```

18.13.1.2. Response

Example 18.68. Create share type: JSON response

```
{
  "volume_type": {
    "os-share-type-access:is_public": true,
    "required_extra_specs": {
      "driver_handles_share_servers": true
    },
    "extra_specs": {
      "snapshot_support": "True",
      "driver_handles_share_servers": "True"
    },
    "name": "my_new_volume_type",
    "id": "1d600d02-26a7-4b23-af3d-7d51860fe858"
  },
  "share_type": {
    "os-share-type-access:is_public": true,
    "required_extra_specs": {
```

```
        "driver_handles_share_servers": true
    },
    "extra_specs": {
        "snapshot_support": "True",
        "driver_handles_share_servers": "True"
    },
    "name": "my_new_volume_type",
    "id": "1d600d02-26a7-4b23-af3d-7d51860fe858"
}
```

18.13.2. List share types

Method	URI	Description
GET	/v2/{tenant_id}/types	Lists all share types.

Normal response codes: 200

18.13.2.1. Request

This table shows the header parameters for the list share types request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list share types request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.13.2.2. Response

Example 18.69. List share types: JSON response

```
{
  "volume_types": [
    {
      "os-share-type-access:is_public": true,
      "required_extra_specs": {
        "driver_handles_share_servers": "True"
      },
      "extra_specs": {
        "snapshot_support": "True",
        "driver_handles_share_servers": "True"
      },
      "name": "default",
      "id": "be27425c-f807-4500-a056-d00721db45cf"
    },
    {
      "os-share-type-access:is_public": true,
      "required_extra_specs": {
        "driver_handles_share_servers": "false"
      },
      "extra_specs": {
        "snapshot_support": "True",
        "driver_handles_share_servers": "false"
      },
      "name": "d",
      "id": "f015bebe-c38b-4c49-8832-00143b10253b"
    }
  ],
  "share_types": [
```

```
{
  {
    "os-share-type-access:is_public": true,
    "required_extra_specs": {
      "driver_handles_share_servers": "True"
    },
    "extra_specs": {
      "snapshot_support": "True",
      "driver_handles_share_servers": "True"
    },
    "name": "default",
    "id": "be27425c-f807-4500-a056-d00721db45cf"
  },
  {
    "os-share-type-access:is_public": true,
    "required_extra_specs": {
      "driver_handles_share_servers": "false"
    },
    "extra_specs": {
      "snapshot_support": "True",
      "driver_handles_share_servers": "false"
    },
    "name": "d",
    "id": "f015bebe-c38b-4c49-8832-00143b10253b"
  }
}
```

18.13.3. List default share types

Method	URI	Description
GET	/v2/{tenant_id}/types/default	Lists default share types.

Normal response codes: 200

18.13.3.1. Request

This table shows the header parameters for the list default share types request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list default share types request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.13.3.2. Response

Example 18.70. List default share types: JSON response

```
{
  "volume_type": {
    "required_extra_specs": null,
    "extra_specs": {
      "snapshot_support": "True",
      "driver_handles_share_servers": "True"
    },
    "name": "default",
    "id": "be27425c-f807-4500-a056-d00721db45cf"
  },
  "share_type": {
    "required_extra_specs": null,
    "extra_specs": {
      "snapshot_support": "True",
      "driver_handles_share_servers": "True"
    },
    "name": "default",
    "id": "be27425c-f807-4500-a056-d00721db45cf"
  }
}
```


18.13.4. Delete share type

Method	URI	Description
DELETE	/v2/{tenant_id}/types/{share_type_id}	Deletes a share type.

Normal response codes: 202

18.13.4.1. Request

This table shows the header parameters for the delete share type request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the delete share type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_type_id}	UUID	The UUID of the share type.

This operation does not accept a request body.

18.13.5. List extra specs

Method	URI	Description
GET	/v2/{tenant_id}/types/{share_type_id}/extra_specs	Lists the extra specifications for a share type.

Normal response codes: 200

18.13.5.1. Request

This table shows the header parameters for the list extra specs request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list extra specs request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_type_id}	UUID	The UUID of the share type.

This operation does not accept a request body.

18.13.5.2. Response

Example 18.71. List extra specs: JSON response

```
{
  "extra_specs": {
    "snapshot_support": "True",
    "driver_handles_share_servers": "True"
  }
}
```

18.13.6. Set extra spec for share type

Method	URI	Description
POST	/v2/{tenant_id}/types/{share_type_id}/extra_specs	Sets an extra specification for the share type.

Each driver implementation determines which extra specification keys it uses. For details, see [Capabilities and Extra-Specs](#) and documentation for your driver.

Normal response codes: 200

18.13.6.1. Request

This table shows the header parameters for the set extra spec for share type request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the set extra spec for share type request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_type_id}	UUID	The UUID of the share type.

Example 18.72. Set extra spec for share type: JSON request

```
{
  "extra_specs": {
    "my_key": "my_value"
  }
}
```

18.13.6.2. Response

Example 18.73. Set extra spec for share type: JSON response

```
{
  "extra_specs": {
    "my_key": "my_value"
  }
}
```

18.13.7. Unset an extra spec

Method	URI	Description
DELETE	/v2/{tenant_id}/types/{share_type_id}/extra_specs/{key}	Unsets an extra specification for the share type.

Normal response codes: 202

18.13.7.1. Request

This table shows the header parameters for the unset an extra spec request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the unset an extra spec request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_type_id}	UUID	The UUID of the share type.
{extra-spec-key}	String	The extra specification key.

This operation does not accept a request body.

18.13.8. Add share type access

Method	URI	Description
POST	/v2/{tenant_id}/types/{share_type_id}/action	Adds share type access for a project.

You can add access to private share types only.

Normal response codes: 202

18.13.8.1. Request

This table shows the header parameters for the add share type access request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the add share type access request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_type_id}	UUID	The UUID of the share type.

Example 18.74. Add share type access: JSON request

```
{
  "addProjectAccess": {
    "project": "e1284adea3ee4d2482af5ed214f3ad90"
  }
}
```

18.13.9. Remove share type access

Method	URI	Description
POST	/v2/{tenant_id}/types/{share_type_id}/action	Removes share type access from a project.

You can remove access from private share types only.

Normal response codes: 202

18.13.9.1. Request

This table shows the header parameters for the remove share type access request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the remove share type access request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_type_id}	UUID	The UUID of the share type.

Example 18.75. Remove share type access: JSON request

```
{
  "removeProjectAccess": {
    "project": "818a3f48dcd644909b3fa2e45a399a27"
  }
}
```

18.13.10. Show share type access details

Method	URI	Description
GET	/v2/{tenant_id}/types/{share_type_id}/share_type_access	Shows access details for a share type.

You can view access details for private share types only.

Normal response codes: 200

18.13.10.1. Request

This table shows the header parameters for the show share type access details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show share type access details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_type_id}	UUID	The UUID of the share type.

This operation does not accept a request body.

18.13.10.2. Response

Example 18.76. Show share type access details: JSON response

```
{
  "share_type_access": [
    {
      "share_type_id": "1732f284-401d-41d9-a494-425451e8b4b8",
      "project_id": "818a3f48dcd644909b3fa2e45a399a27"
    },
    {
      "share_type_id": "1732f284-401d-41d9-a494-425451e8b4b8",
      "project_id": "e1284adea3ee4d2482af5ed214f3ad90"
    }
  ]
}
```

18.14. Back-end storage pools

An administrator can list all back-end storage pools that are known to the scheduler service.

Method	URI	Description
GET	/v2/{tenant_id}/scheduler-stats/pools	Lists all back-end storage pools.

Method	URI	Description
GET	/v2/{tenant_id}/scheduler-stats/pools/detail	Lists all storage pools for a back end, with details.

18.14.1. List back-end storage pools

Method	URI	Description
GET	/v2/{tenant_id}/scheduler-stats/pools	Lists all back-end storage pools.

Normal response codes: 200

18.14.1.1. Request

This table shows the header parameters for the list back-end storage pools request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list back-end storage pools request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.14.1.2. Response

Example 18.77. List back-end storage pools: JSON response

```
{
  "pools": [
    {
      "host": "manila2",
      "name": "manila2@generic1#GENERIC1",
      "pool": "GENERIC1",
      "backend": "generic1"
    },
    {
      "host": "manila2",
      "name": "manila2@unmanage1#UNMANAGE1",
      "pool": "UNMANAGE1",
      "backend": "unmanage1"
    },
    {
      "host": "manila2",
      "name": "manila2@ams_backend#AMS_BACKEND",
      "pool": "AMS_BACKEND",
      "backend": "ams_backend"
    }
  ]
}
```

18.14.2. List back-end storage pools with details

Method	URI	Description
GET	/v2/{tenant_id}/scheduler-stats/pools/detail	Lists all storage pools for a back end, with details.

Normal response codes: 200

18.14.2.1. Request

This table shows the header parameters for the list back-end storage pools with details request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list back-end storage pools with details request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.14.2.2. Response

Example 18.78. List back-end storage pools with details: JSON response

```
{
  "pools": [
    {
      "pool": "LONDON",
      "host": "nosb-devstack",
      "name": "nosb-devstack@london#LONDON",
      "capabilities": {
        "qos": false,
        "consistency_group_support": "pool",
        "timestamp": "2015-09-21T08:58:56.190856",
        "share_backend_name": "LONDON",
        "server_pools_mapping": {
          "1320689d-80f4-49f6-8a70-0e2c1ed8ad90": [],
          "3a4caac5-0880-4629-a334-6cdda88a0c0e": []
        },
      },
      "driver_handles_share_servers": true,
      "driver_version": "1.0",
      "total_capacity_gb": "unknown",
      "reserved_percentage": 0,
      "pools": null,
      "vendor_name": "Open Source",
      "snapshot_support": true,
      "free_capacity_gb": "unknown",
      "storage_protocol": "NFS_CIFS"
    }
  ]
}
```

```
    },  
    "backend": "london"  
  }  
]  
}
```

18.15. Services

Lists services. Services include manila-share and manila-scheduler and their binaries, hosts, availability zones, current statuses, and states (up or down).

Administrators can also enable or disable a service.



Caution

For old microversions 1.0-2.6 replace URL part services with os-services.

Method	URI	Description
GET	/v2/{tenant_id}/services	Lists all services.
PUT	/v2/{tenant_id}/services/enable	Enables a service.
PUT	/v2/{tenant_id}/services/disable	Disables a service.

18.15.1. List services

Method	URI	Description
GET	/v2/{tenant_id}/services	Lists all services.

Normal response codes: 200

18.15.1.1. Request

This table shows the header parameters for the list services request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list services request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.15.1.2. Response

Example 18.79. List services: JSON response

```
{
  "services": [
    {
      "status": "enabled",
      "binary": "manila-share",
      "zone": "nova",
      "host": "manila2@generic1",
      "updated_at": "2015-09-07T13:03:57.000000",
      "state": "up",
      "id": 1
    },
    {
      "status": "enabled",
      "binary": "manila-scheduler",
      "zone": "nova",
      "host": "manila2",
      "updated_at": "2015-09-07T13:03:57.000000",
      "state": "up",
      "id": 2
    }
  ]
}
```

18.15.2. Enable service

Method	URI	Description
PUT	/v2/{tenant_id}/services/enable	Enables a service.

Normal response codes: 200

18.15.2.1. Request

This table shows the header parameters for the enable service request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the enable service request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 18.80. Enable service: JSON request

```
{
  "binary": "manila-share",
  "host": "openstack@cmode"
}
```

18.15.2.2. Response

Example 18.81. Enable service: JSON response

```
{
  "disabled": false,
  "binary": "manila-share",
  "host": "openstack@cmode"
}
```

18.15.3. Disable service

Method	URI	Description
PUT	/v2/{tenant_id}/services/disable	Disables a service.

Normal response codes: 200

18.15.3.1. Request

This table shows the header parameters for the disable service request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the disable service request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 18.82. Disable service: JSON request

```
{
  "binary": "manila-share",
  "host": "openstack@cmode"
}
```

18.15.3.2. Response

Example 18.83. Disable service: JSON response

```
{
  "disabled": true,
  "binary": "manila-share",
  "host": "openstack@cmode"
}
```

18.16. Availability zones

Describes availability zones.



Caution

For old microversions 1.0-2.6 replace URL part availability-zones with os-availability-zone.

Method	URI	Description
GET	/v2/{tenant_id}/availability-zones	Lists all availability zones.

18.16.1. List availability zones

Method	URI	Description
GET	/v2/{tenant_id}/availability-zones	Lists all availability zones.

Normal response codes: 200

18.16.1.1. Request

This table shows the header parameters for the list availability zones request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list availability zones request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

18.16.1.2. Response

Example 18.84. List availability zones: JSON response

```
{
  "availability_zones": [
    {
      "name": "nova",
      "created_at": "2015-09-18T09:50:55.000000",
      "updated_at": null,
      "id": "388c983d-258e-4a0e-b1ba-10da37d766db"
    }
  ]
}
```

18.17. Manage share

Configures Shared File Systems to manage or unmanage a share.

The share unmanage operation is not supported for shares that are created on top of share servers (created with share networks).

You can unmanage a share that has no dependent snapshots.

These APIs are available for micro-versions 1.0-2.6

Starting with micro-version 2.7 these APIs are available under 'shares' API group.

Method	URI	Description
POST	/v2/{tenant_id}/os-share-manage	Configures Shared File Systems to manage a share.

Method	URI	Description
POST	/v2/{tenant_id}/os-share-unmanage/{share_id}/unmanage	Configures Shared File Systems to stop managing a share.

18.17.1. Manage share

Method	URI	Description
POST	/v2/{tenant_id}/os-share-manage	Configures Shared File Systems to manage a share.

Normal response codes: 200

18.17.1.1. Request

This table shows the header parameters for the manage share request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the manage share request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 18.85. Manage share: JSON request

```
{
  "share": {
    "protocol": "nfs",
    "name": "share_texas1",
    "share_type": "d",
    "is_public": false,
    "driver_options": {
      "opt1": "opt1",
      "opt2": "opt2"
    },
    "export_path": "10.254.0.5:/shares/share-42033c24-0261-424f-abda-4fef2f6dbfd5",
    "service_host": "manila2@unmanage1#UNMANAGE1",
    "description": "Lets manage share."
  }
}
```

18.17.1.2. Response

Example 18.86. Manage share: JSON response

```
{
  "share": {
    "links": [
      {
        "href": "http://172.18.198.54:8786/v2/16e1ab15c35a457e9c2b2aa189f544e1/shares/00137b40-ca06-4ae8-83a3-2c5989eebcce",
        "rel": "self"
      },
      {
        "href": "http://172.18.198.54:8786/16e1ab15c35a457e9c2b2aa189f544e1/shares/00137b40-ca06-4ae8-83a3-2c5989eebcce",

```

```

        "rel": "bookmark"
      }
    ],
    "availability_zone": null,
    "share_network_id": null,
    "export_locations": [],
    "share_server_id": null,
    "snapshot_id": null,
    "id": "00137b40-ca06-4ae8-83a3-2c5989eebcce",
    "size": null,
    "share_type": "14747856-08e5-494f-ab40-a64b9d20d8f7",
    "share_type_name": "d",
    "export_location": "10.254.0.5:/shares/share-42033c24-0261-424f-abda-4fef2f6dbfd5",
    "consistency_group_id": null,
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "metadata": {},
    "status": "manage_starting",
    "description": "Lets manage share.",
    "host": "manila2@unmanage1#UNMANAGE1",
    "access_rules_status": "active",
    "has_replicas": false,
    "replication_type": null,
    "is_public": false,
    "snapshot_support": true,
    "name": "share_texas1",
    "created_at": "2015-09-17T16:21:12.000000",
    "share_proto": "NFS",
    "volume_type": "d",
    "source_cgsnapshot_member_id": null
  }
}

```

18.17.2. Unmanage share

Method	URI	Description
POST	/v2/{tenant_id}/os-share-unmanage/{share_id}/unmanage	Configures Shared File Systems to stop managing a share.

Normal response codes: 202

18.17.2.1. Request

This table shows the header parameters for the unmanage share request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the unmanage share request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{share_id}	UUID	The UUID of the share.

This operation does not accept a request body.

18.18. Quota sets

Provides quotas management support.



Caution

For old microversions 1.0-2.6 replace URL part quota-sets with os-quota-sets.

Method	URI	Description
GET	/v2/{tenant_id}/quota-sets/{tenant_id}?user_id	Shows quotas for a tenant.
PUT	/v2/{tenant_id}/quota-sets/{tenant_id}?user_id	Updates quotas for a tenant.
DELETE	/v2/{tenant_id}/quota-sets/{tenant_id}?user_id	Deletes quotas for a tenant. The quota reverts to the default quota.
GET	/v2/{tenant_id}/quota-sets/{tenant_id}/defaults	Shows default quotas for a tenant.

18.18.1. Show quotas

Method	URI	Description
GET	/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}	Shows quotas for a tenant.

If you specify the optional `user_id` query parameter, you get the quotas for this user in the tenant. If you omit this parameter, you get the quotas for the project.

Normal response codes: 200

18.18.1.1. Request

This table shows the header parameters for the show quotas request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{tenant_id}	UUID	The UUID for the tenant for which you want to show, update, or delete quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the administrative tenant.

This operation does not accept a request body.

18.18.1.2. Response

Example 18.87. Show quotas: JSON response

```
{
  "quota_set": {
    "gigabytes": 1000,
    "shares": 50,
    "snapshot_gigabytes": 1000,
    "snapshots": 50,
    "id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "share_networks": 10
  }
}
```

18.18.2. Update quotas

Method	URI	Description
PUT	/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}	Updates quotas for a tenant.

If you specify the optional `user_id` query parameter, you update the quotas for this user in the tenant. If you omit this parameter, you update the quotas for the project.

Normal response codes: 200

18.18.2.1. Request

This table shows the header parameters for the update quotas request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the update quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{tenant_id}	UUID	The UUID for the tenant for which you want to show, update, or delete quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the administrative tenant.

Example 18.88. Update quotas: JSON request

```
{
  "quota_set": {
    "tenant_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "snapshot_gigabytes": 999,
    "snapshots": 49,
    "share_networks": 9
  }
}
```

18.18.2.2. Response

Example 18.89. Update quotas: JSON response

```
{
  "quota_set": {
    "gigabytes": 1000,
    "snapshot_gigabytes": 999,
    "shares": 50,
    "snapshots": 49,
    "share_networks": 9
  }
}
```

18.18.3. Delete quotas

Method	URI	Description
DELETE	/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}	Deletes quotas for a tenant. The quota reverts to the default quota.

If you specify the optional `user_id` query parameter, you delete the quotas for this user in the tenant. If you omit this parameter, you delete the quotas for the project.

Normal response codes: 202

18.18.3.1. Request

This table shows the header parameters for the delete quotas request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the delete quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{tenant_id}	UUID	The UUID for the tenant for which you want to show, update, or delete quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the administrative tenant.

This operation does not accept a request body.

18.18.4. Show default quotas

Method	URI	Description
GET	/v2/{tenant_id}/quota-sets/ {tenant_id}/defaults	Shows default quotas for a tenant.

Normal response codes: 200

18.18.4.1. Request

This table shows the header parameters for the show default quotas request:

Name	Type	Description
X-Openstack-Manila-API-Version	String (Optional)	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-API-Version: 2.15". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show default quotas request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{tenant_id}	UUID	The UUID for the tenant for which you want to show, update, or delete quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the administrative tenant.

This operation does not accept a request body.

18.18.4.2. Response

Example 18.90. Show default quotas: JSON response

```
{
  "quota_set": {
    "gigabytes": 1000,
    "shares": 50,
    "snapshot_gigabytes": 1000,
    "snapshots": 50,
    "id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "share_networks": 10
  }
}
```

19. Shared File Systems API v1 (SUPPORTED)

Provides coordinated access to shared or distributed file systems.

The Shared File Systems API v1 is functionally identical to the [Shared File Systems API v2](#). Subsequent API v2 micro-versions, such as v2.1, differ from API v1.

(Since API v2.0) The API uses Compute-style micro-versions.

20. Telemetry API v2 (CURRENT)

Manages alarms, meters, samples, resources, and capabilities through a set of services:

- [Aodh](#). An alarm service.
- [Ceilometer](#). A data collection service.
- [Gnocchi](#). A time-series database and resource indexing service.

If Gnocchi is enabled, meters, samples, and resources API operations return the 410 response code.

Method	URI	Description
Alarms		
GET	/v2/alarms{?q}	Lists alarms, based on a query.
POST	/v2/alarms{?data}	Creates an alarm.
GET	/v2/alarms/{alarm_id}	Shows details for an alarm, by alarm ID.
PUT	/v2/alarms/{alarm_id}{?data}	Updates an alarm.
DELETE	/v2/alarms/{alarm_id}	Deletes an alarm, by alarm ID.
PUT	/v2/alarms/{alarm_id}/state{?state}	Sets the state of an alarm.
GET	/v2/alarms/{alarm_id}/state	Shows the state for an alarm, by alarm ID.
GET	/v2/alarms/{alarm_id}/history{?q}	Assembles and shows the history for an alarm, by alarm ID.
Meters		
GET	/v2/meters{?q, limit}	Lists meters, based on the data recorded so far.
POST	/v2/meters/{meter_name}{?direct, samples}	Adds samples to a meter, by meter name.
GET	/v2/meters/{meter_name}{?q, limit}	Lists samples for a meter, by meter name.
GET	/v2/meters/{meter_name}/statistics{?q, groupby, period, aggregate, limit}	Computes and lists statistics for samples in a time range.
Samples		
GET	/v2/samples{?q, limit}	Lists all known samples, based on the data recorded so far.
GET	/v2/samples/{sample_id}	Shows details for a sample, by sample ID.
Resources		
GET	/v2/resources{?q, meter_links}	Lists definitions for all resources.
GET	/v2/resources/{resource_id}	Shows details for a resource, by resource ID.
Capabilities		
GET	/v2/capabilities	A representation of the API and storage capabilities. Usually, the storage driver imposes constraints.
Events		
GET	/v2/events{?q, limit}	Lists all events.
GET	/v2/events/{message_id}	Shows details for an event.

20.1. Alarms

Lists, creates, gets details for, updates, and deletes alarms.

Method	URI	Description
GET	/v2/alarms{?q}	Lists alarms, based on a query.
POST	/v2/alarms{?data}	Creates an alarm.
GET	/v2/alarms/{alarm_id}	Shows details for an alarm, by alarm ID.
PUT	/v2/alarms/{alarm_id}{?data}	Updates an alarm.
DELETE	/v2/alarms/{alarm_id}	Deletes an alarm, by alarm ID.
PUT	/v2/alarms/{alarm_id}/state{?state}	Sets the state of an alarm.
GET	/v2/alarms/{alarm_id}/state	Shows the state for an alarm, by alarm ID.
GET	/v2/alarms/{alarm_id}/history{?q}	Assembles and shows the history for an alarm, by alarm ID.

20.1.1. List alarms

Method	URI	Description
GET	/v2/alarms{?q}	Lists alarms, based on a query.

Normal response codes: 200

20.1.1.1. Request

This table shows the query parameters for the list alarms request:

Name	Type	Description
q	List (Optional)	Filters the response by one or more arguments. For example: ?q.field=Foo&q.value=my_text.

This operation does not accept a request body.

20.1.1.2. Response

Example 20.1. List alarms: JSON response

```
[
  {
    "alarm_actions": [
      "http://site:8000/alarm"
    ],
    "alarm_id": null,
    "combination_rule": null,
    "description": "An alarm",
    "enabled": true,
    "insufficient_data_actions": [
      "http://site:8000/nodata"
    ],
    "name": "SwiftObjectAlarm",
    "ok_actions": [
      "http://site:8000/ok"
    ],
    "project_id": "c96c887c216949acbd8b494863567",
    "repeat_actions": false,
    "state": "ok",
    "state_timestamp": "2013-11-21T12:33:08.486228",
    "threshold_rule": null,
    "timestamp": "2013-11-21T12:33:08.486221",
    "type": "threshold",
    "user_id": "c96c887c216949acbd8b494863567"
  }
]
```

Example 20.2. List alarms: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
  <value>
    <alarm_actions>
```

```
<item>http://site:8000/alarm</item>
</alarm_actions>
<alarm_id nil="true" />
<combination_rule nil="true" />
<description>An alarm</description>
<enabled>true</enabled>
<insufficient_data_actions>
  <item>http://site:8000/nodata</item>
</insufficient_data_actions>
<name>SwiftObjectAlarm</name>
<ok_actions>
  <item>http://site:8000/ok</item>
</ok_actions>
<project_id>c96c887c216949acbd8b494863567</project_id>
<repeat_actions>false</repeat_actions>
<state>ok</state>
<state_timestamp>2013-11-21T12:33:08.486228</state_timestamp>
<threshold_rule nil="true" />
<timestamp>2013-11-21T12:33:08.486221</timestamp>
<type>threshold</type>
<user_id>c96c887c216949acbd8b494863567</user_id>
</value>
</values>
```

This operation does not return a response body.

20.1.2. Create alarm

Method	URI	Description
POST	/v2/alarms{?data}	Creates an alarm.

Normal response codes: 200

20.1.2.1. Request

This table shows the query parameters for the create alarm request:

Name	Type	Description
data	Alarm (Optional)	An alarm within the request body.

This operation does not accept a request body.

20.1.2.2. Response

Example 20.3. Create alarm: JSON response

```
{
  "alarm_actions": [
    "http://site:8000/alarm"
  ],
  "alarm_id": null,
  "combination_rule": null,
  "description": "An alarm",
  "enabled": true,
  "insufficient_data_actions": [
    "http://site:8000/nodata"
  ],
  "name": "SwiftObjectAlarm",
  "ok_actions": [
    "http://site:8000/ok"
  ],
  "project_id": "c96c887c216949acbd8b494863567",
  "repeat_actions": false,
  "state": "ok",
  "state_timestamp": "2013-11-21T12:33:08.486228",
  "threshold_rule": null,
  "timestamp": "2013-11-21T12:33:08.486221",
  "type": "threshold",
  "user_id": "c96c887c216949acbd8b494863567"
}
```

Example 20.4. Create alarm: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
  <alarm_actions>
    <item>http://site:8000/alarm</item>
  </alarm_actions>
  <alarm_id nil="true" />
</value>
```

```
<combination_rule nil="true" />
<description>An alarm</description>
<enabled>true</enabled>
<insufficient_data_actions>
  <item>http://site:8000/nodata</item>
</insufficient_data_actions>
<name>SwiftObjectAlarm</name>
<ok_actions>
  <item>http://site:8000/ok</item>
</ok_actions>
<project_id>c96c887c216949acbd8b494863567</project_id>
<repeat_actions>false</repeat_actions>
<state>ok</state>
<state_timestamp>2013-11-21T12:33:08.486228</state_timestamp>
<threshold_rule nil="true" />
<timestamp>2013-11-21T12:33:08.486221</timestamp>
<type>threshold</type>
<user_id>c96c887c216949acbd8b494863567</user_id>
</value>
```

This operation does not return a response body.

20.1.3. Show alarm details

Method	URI	Description
GET	/v2/alarms/{alarm_id}	Shows details for an alarm, by alarm ID.

Normal response codes: 200

20.1.3.1. Request

This table shows the URI parameters for the show alarm details request:

Name	Type	Description
{alarm_id}	UUID	The UUID of the alarm.

This operation does not accept a request body.

20.1.3.2. Response

Example 20.5. Show alarm details: JSON response

```
{
  "alarm_actions": [
    "http://site:8000/alarm"
  ],
  "alarm_id": null,
  "combination_rule": null,
  "description": "An alarm",
  "enabled": true,
  "insufficient_data_actions": [
    "http://site:8000/nodata"
  ],
  "name": "SwiftObjectAlarm",
  "ok_actions": [
    "http://site:8000/ok"
  ],
  "project_id": "c96c887c216949acbd8b494863567",
  "repeat_actions": false,
  "state": "ok",
  "state_timestamp": "2013-11-21T12:33:08.486228",
  "threshold_rule": null,
  "timestamp": "2013-11-21T12:33:08.486221",
  "type": "threshold",
  "user_id": "c96c887c216949acbd8b494863567"
}
```

Example 20.6. Show alarm details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
  <alarm_actions>
    <item>http://site:8000/alarm</item>
  </alarm_actions>
  <alarm_id nil="true" />
  <combination_rule nil="true" />
  <description>An alarm</description>
```

```
<enabled>true</enabled>
<insufficient_data_actions>
  <item>http://site:8000/nodata</item>
</insufficient_data_actions>
<name>SwiftObjectAlarm</name>
<ok_actions>
  <item>http://site:8000/ok</item>
</ok_actions>
<project_id>c96c887c216949acbd8b494863567</project_id>
<repeat_actions>false</repeat_actions>
<state>ok</state>
<state_timestamp>2013-11-21T12:33:08.486228</state_timestamp>
<threshold_rule nil="true" />
<timestamp>2013-11-21T12:33:08.486221</timestamp>
<type>threshold</type>
<user_id>c96c887c216949acbd8b494863567</user_id>
</value>
```

This operation does not return a response body.

20.1.4. Update alarm

Method	URI	Description
PUT	/v2/alarms/{alarm_id}{?data}	Updates an alarm.

Normal response codes: 200

20.1.4.1. Request

This table shows the URI parameters for the update alarm request:

Name	Type	Description
{alarm_id}	UUID	The UUID of the alarm.

This table shows the query parameters for the update alarm request:

Name	Type	Description
data	Alarm (Optional)	An alarm within the request body.

This operation does not accept a request body.

20.1.4.2. Response

Example 20.7. Update alarm: JSON response

```
{
  "alarm_actions": [
    "http://site:8000/alarm"
  ],
  "alarm_id": null,
  "combination_rule": null,
  "description": "An alarm",
  "enabled": true,
  "insufficient_data_actions": [
    "http://site:8000/nodata"
  ],
  "name": "SwiftObjectAlarm",
  "ok_actions": [
    "http://site:8000/ok"
  ],
  "project_id": "c96c887c216949acbdafb8b494863567",
  "repeat_actions": false,
  "state": "ok",
  "state_timestamp": "2013-11-21T12:33:08.486228",
  "threshold_rule": null,
  "timestamp": "2013-11-21T12:33:08.486221",
  "type": "threshold",
  "user_id": "c96c887c216949acbdafb8b494863567"
}
```

Example 20.8. Update alarm: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<value>
  <alarm_actions>
    <item>http://site:8000/alarm</item>
  </alarm_actions>
  <alarm_id nil="true" />
  <combination_rule nil="true" />
  <description>An alarm</description>
  <enabled>true</enabled>
  <insufficient_data_actions>
    <item>http://site:8000/nodata</item>
  </insufficient_data_actions>
  <name>SwiftObjectAlarm</name>
  <ok_actions>
    <item>http://site:8000/ok</item>
  </ok_actions>
  <project_id>c96c887c216949acbd8b494863567</project_id>
  <repeat_actions>false</repeat_actions>
  <state>ok</state>
  <state_timestamp>2013-11-21T12:33:08.486228</state_timestamp>
  <threshold_rule nil="true" />
  <timestamp>2013-11-21T12:33:08.486221</timestamp>
  <type>threshold</type>
  <user_id>c96c887c216949acbd8b494863567</user_id>
</value>
```

This operation does not return a response body.

20.1.5. Delete alarm

Method	URI	Description
DELETE	/v2/alarms/{alarm_id}	Deletes an alarm, by alarm ID.

Normal response codes: 204

20.1.5.1. Request

This table shows the URI parameters for the delete alarm request:

Name	Type	Description
{alarm_id}	UUID	The UUID of the alarm.

This operation does not accept a request body.

20.1.6. Update alarm state

Method	URI	Description
PUT	/v2/alarms/{alarm_id}/state{?state}	Sets the state of an alarm.

Normal response codes: 200

20.1.6.1. Request

This table shows the URI parameters for the update alarm state request:

Name	Type	Description
{alarm_id}	UUID	The UUID of the alarm.

This table shows the query parameters for the update alarm state request:

Name	Type	Description
state	String (Required)	The alarm state. A valid value is ok, alarm, or insufficient data.

This operation does not accept a request body.

20.1.7. Show alarm state

Method	URI	Description
GET	/v2/alarms/{alarm_id}/state	Shows the state for an alarm, by alarm ID.

Normal response codes: 200

20.1.7.1. Request

This table shows the URI parameters for the show alarm state request:

Name	Type	Description
{alarm_id}	UUID	The UUID of the alarm.

20.1.8. Show alarm history

Method	URI	Description
GET	/v2/alarms/{alarm_id}/history{?q}	Assembles and shows the history for an alarm, by alarm ID.

Normal response codes: 200

20.1.8.1. Request

This table shows the URI parameters for the show alarm history request:

Name	Type	Description
{alarm_id}	UUID	The UUID of the alarm.

This table shows the query parameters for the show alarm history request:

Name	Type	Description
q	List (Optional)	Filters the response by one or more arguments. For example: ?q.field=Foo&q.value=my_text.

This operation does not accept a request body.

20.2. Meters

Lists all meters, adds samples to meters, and lists samples for meters. For list operations, if you do not explicitly set the `limit` query parameter, a default limit is applied. The default limit is the `default_api_return_limit` configuration option value.

Also, computes and lists statistics for samples in a time range. You can use the `aggregate` query parameter in the `statistics` URI to explicitly select the `stddev`, `cardinality`, or any other standard function. For example:

```
GET /v2/meters/METER_NAME/statistics?aggregate.func=NAME&aggregate.param=VALUE
```

The `aggregate.param` parameter value is optional for all functions except the `cardinality` function.

The API silently ignores any duplicate `aggregate` function and parameter pairs.

The API accepts and storage drivers support duplicate functions with different parameter values. In this example, the `cardinality` function is accepted twice with two different parameter values:

```
GET /v2/meters/METER_NAME/statistics?aggregate.func=cardinality&aggregate.  
param=resource_id  
&aggregate.func=cardinality&aggregate.param=project_id
```

Examples:

Use the `stddev` function to request the standard deviation of CPU utilization:

```
GET /v2/meters/cpu_util/statistics?aggregate.func=stddev
```

The response looks like this:

```
[
  {
    "aggregate": {
      "stddev": 0.6858829
    },
    "duration_start": "2014-01-30T11:13:23",
    "duration_end": "2014-01-31T16:07:13",
    "duration": 104030,
    "period": 0,
    "period_start": "2014-01-30T11:13:23",
    "period_end": "2014-01-31T16:07:13",
    "groupby": null,
    "unit": "%"
  }
]
```

Use the `cardinality` function with the project ID to return the number of distinct tenants with images:

```
GET /v2/meters/image/statistics?aggregate.func=cardinality&aggregate.param=project_id
```

The following, more complex, example determines:

- The number of distinct instances (`cardinality`)
- The total number of instance samples (`count`) for a tenant in 15-minute intervals (`period` and `groupby` options)

```
GET /v2/meters/instance/statistics?aggregate.func=cardinality&aggregate.param=resource_id
    &aggregate.func=count&groupby=project_id&period=900
```

The response looks like this:

```
[
  {
    "count": 19,
    "aggregate": {
      "count": 19,
      "cardinality/resource_id": 3
    },
    "duration": 328.47803,
    "duration_start": "2014-01-31T10:00:41.823919",
    "duration_end": "2014-01-31T10:06:10.301948",
    "period": 900,
    "period_start": "2014-01-31T10:00:00",
    "period_end": "2014-01-31T10:15:00",
    "groupby": {
      "project_id": "061a5c91811e4044b7dc86c6136c4f99"
    },
    "unit": "instance"
  },
  {
    "count": 22,
    "aggregate": {
```

```

        "count": 22,
        "cardinality/resource_id": 4
      },
      "duration": 808.00385,
      "duration_start": "2014-01-31T10:15:15",
      "duration_end": "2014-01-31T10:28:43.003840",
      "period": 900,
      "period_start": "2014-01-31T10:15:00",
      "period_end": "2014-01-31T10:30:00",
      "groupby": {
        "project_id": "061a5c91811e4044b7dc86c6136c4f99"
      },
      "unit": "instance"
    },
    {
      "count": 2,
      "aggregate": {
        "count": 2,
        "cardinality/resource_id": 2
      },
      "duration": 0,
      "duration_start": "2014-01-31T10:35:15",
      "duration_end": "2014-01-31T10:35:15",
      "period": 900,
      "period_start": "2014-01-31T10:30:00",
      "period_end": "2014-01-31T10:45:00",
      "groupby": {
        "project_id": "061a5c91811e4044b7dc86c6136c4f99"
      },
      "unit": "instance"
    }
  ]

```

Method	URI	Description
GET	/v2/meters{?q,limit}	Lists meters, based on the data recorded so far.
POST	/v2/meters/{meter_name}{?direct,samples}	Adds samples to a meter, by meter name.
GET	/v2/meters/{meter_name}{?q,limit}	Lists samples for a meter, by meter name.
GET	/v2/meters/{meter_name}/statistics{?q,groupby,period,aggregate,limit}	Computes and lists statistics for samples in a time range.

20.2.1. List meters

Method	URI	Description
GET	/v2/meters{?q,limit}	Lists meters, based on the data recorded so far.

Normal response codes: 200

20.2.1.1. Request

This table shows the query parameters for the list meters request:

Name	Type	Description
q	List (Optional)	Filters the response by one or more arguments. For example: ?q.field=Foo&q.value=my_text.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

20.2.1.2. Response

Example 20.9. List meters: JSON response

```
[
  {
    "meter_id":
    "YmQ5NDMxYzEtOGQ2OS00YWQzLTgwM2EtOGQ0YTZiODlmZDM2K2luc3RhbmNl",
    "name": "instance",
    "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
    "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
    "source": "openstack",
    "type": "gauge",
    "unit": "instance",
    "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff"
  }
]
```

Example 20.10. List meters: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
  <value>
    <name>instance</name>
    <type>gauge</type>
    <unit>instance</unit>
    <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
    <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
    <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
    <source>openstack</source>
    <meter_id>YmQ5NDMxYzEtOGQ2OS00YWQzLTgwM2EtOGQ0YTZiODlmZDM2K2luc3RhbmNl</
meter_id>
  </value>
```

```
</values>
```

This operation does not return a response body.

20.2.2. Add samples to meter

Method	URI	Description
POST	/v2/meters/{meter_name}{?direct,samples}	Adds samples to a meter, by meter name.

If you attempt to add a sample that is not supported, this call returns the 409 response code.

Normal response codes: 200

Error response codes: conflict (409)

20.2.2.1. Request

This table shows the URI parameters for the add samples to meter request:

Name	Type	Description
{meter_name}	String	The name of the meter.

This table shows the query parameters for the add samples to meter request:

Name	Type	Description
direct	String (Optional)	Indicates whether the samples are POST ed directly to storage. Set ?direct=True to POST the samples directly to storage.
samples	List (Optional)	A list of samples.

Example 20.11. Add samples to meter: JSON request

```
{
  "id": "8db08c68-bc70-11e4-a8c4-fa163e1d1a9b",
  "metadata": {
    "name1": "value1",
    "name2": "value2"
  },
  "meter": "instance",
  "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
  "recorded_at": "2015-02-24T22:00:32.747930",
  "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
  "source": "openstack",
  "timestamp": "2015-02-24T22:00:32.747930",
  "type": "gauge",
  "unit": "instance",
  "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff",
  "volume": 1.0
}
```

Example 20.12. Add samples to meter: XML request

```
<value>
  <id>8db08c68-bc70-11e4-a8c4-fa163e1d1a9b</id>
  <meter>instance</meter>
```

```

<type>gauge</type>
<unit>instance</unit>
<volume>1.0</volume>
<user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
<project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
<resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
<source>openstack</source>
<timestamp>2015-02-24T22:00:32.747930</timestamp>
<recorded_at>2015-02-24T22:00:32.747930</recorded_at>
<metadata>
  <item>
    <key>name2</key>
    <value>value2</value>
  </item>
  <item>
    <key>name1</key>
    <value>value1</value>
  </item>
</metadata>
</value>

```

This operation does not accept a request body.

20.2.2.2. Response

Example 20.13. Add samples to meter: JSON response

```

{
  "id": "9b23b398-6139-11e5-97e9-bc764e045bf6",
  "metadata": {
    "name1": "value1",
    "name2": "value2"
  },
  "meter": "instance",
  "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
  "recorded_at": "2015-09-22T14:52:54.850725",
  "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
  "source": "openstack",
  "timestamp": "2015-09-22T14:52:54.850718",
  "type": "gauge",
  "unit": "instance",
  "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff",
  "volume": 1
}

```

Example 20.14. Add samples to meter: XML response

```

<?xml version="1.0" encoding="UTF-8"?>
<value>
  <id>9b23b398-6139-11e5-97e9-bc764e045bf6</id>
  <meter>instance</meter>
  <type>gauge</type>
  <unit>instance</unit>
  <volume>1.0</volume>
  <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
  <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
  <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
  <source>openstack</source>
  <timestamp>2015-09-22T14:52:54.850718</timestamp>

```

```
<recorded_at>2015-09-22T14:52:54.850725</recorded_at>
<metadata>
  <item>
    <key>name2</key>
    <value>value2</value>
  </item>
  <item>
    <key>name1</key>
    <value>value1</value>
  </item>
</metadata>
</value>
```

This operation does not return a response body.

20.2.3. List samples for meter

Method	URI	Description
GET	/v2/meters/{meter_name}{?q, limit}	Lists samples for a meter, by meter name.

Normal response codes: 200

20.2.3.1. Request

This table shows the URI parameters for the list samples for meter request:

Name	Type	Description
{meter_name}	String	The name of the meter.

This table shows the query parameters for the list samples for meter request:

Name	Type	Description
q	List (Optional)	Filters the response by one or more arguments. For example: ?q.field=Foo&q.value=my_text.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

20.2.3.2. Response

Example 20.15. List samples for meter: JSON response

```
[
  {
    "id": "9b23b398-6139-11e5-97e9-bc764e045bf6",
    "metadata": {
      "name1": "value1",
      "name2": "value2"
    },
    "meter": "instance",
    "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
    "recorded_at": "2015-09-22T14:52:54.850725",
    "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
    "source": "openstack",
    "timestamp": "2015-09-22T14:52:54.850718",
    "type": "gauge",
    "unit": "instance",
    "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff",
    "volume": 1
  }
]
```

Example 20.16. List samples for meter: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<values>
  <value>
    <id>9b23b398-6139-11e5-97e9-bc764e045bf6</id>
    <meter>instance</meter>
    <type>gauge</type>
    <unit>instance</unit>
    <volume>1.0</volume>
    <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
    <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
    <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
    <source>openstack</source>
    <timestamp>2015-09-22T14:52:54.850718</timestamp>
    <recorded_at>2015-09-22T14:52:54.850725</recorded_at>
    <metadata>
      <item>
        <key>name2</key>
        <value>value2</value>
      </item>
      <item>
        <key>name1</key>
        <value>value1</value>
      </item>
    </metadata>
  </value>
</values>
```

This operation does not return a response body.

20.2.4. Show meter statistics

Method	URI	Description
GET	/v2/meters/{meter_name}/statistics {?q,groupby,period,aggregate,limit}	Computes and lists statistics for samples in a time range.

Normal response codes: 200

20.2.4.1. Request

This table shows the URI parameters for the show meter statistics request:

Name	Type	Description
{meter_name}	String	The name of the meter.

This table shows the query parameters for the show meter statistics request:

Name	Type	Description
q	List (Optional)	Filters the response by one or more arguments. For example: ?q.field=Foo&q.value=my_text.
groupby	Dict (Optional)	Fields for group by aggregation.
period	Int (Optional)	The period, in seconds, for which you want statistics.
aggregate	Dict (Optional)	A list of selectable aggregation functions to apply. For example: <code>GET /v2/meters/METER_NAME/statistics?aggregate.func=cardinality&aggregate.param=resource_id&aggregate.func=cardinality&aggregate.param=project_id</code>
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the <code>limit</code> parameter to make an initial limited request and use the ID of the last-seen item from the response as the <code>marker</code> parameter value in a subsequent limited request.

This operation does not accept a request body.

20.2.4.2. Response

Example 20.17. Show meter statistics: JSON response

```
[
  {
    "avg": 4.5,
    "count": 10,
    "duration": 300,
    "duration_end": "2013-01-04T16:47:00",
    "duration_start": "2013-01-04T16:42:00",
    "max": 9,
    "min": 1,
    "period": 7200,
```



```

        "period_end": "2013-01-04T18:00:00",
        "period_start": "2013-01-04T16:00:00",
        "sum": 45,
        "unit": "GiB"
    }
]

```

Example 20.18. Show meter statistics: XML response

```

<?xml version="1.0" encoding="UTF-8"?>
<values>
  <value>
    <avg>4.5</avg>
    <count>10</count>
    <duration>300.0</duration>
    <duration_end>2013-01-04T16:47:00</duration_end>
    <duration_start>2013-01-04T16:42:00</duration_start>
    <max>9.0</max>
    <min>1.0</min>
    <period>7200</period>
    <period_end>2013-01-04T18:00:00</period_end>
    <period_start>2013-01-04T16:00:00</period_start>
    <sum>45.0</sum>
    <unit>GiB</unit>
  </value>
</values>

```

This operation does not return a response body.

20.3. Samples

Lists all samples and gets information for a sample.

For list operations, if you do not explicitly set the `limit` query parameter, a default limit is applied. The default limit is the `default_api_return_limit` configuration option value.

Method	URI	Description
GET	/v2/samples{?q, limit}	Lists all known samples, based on the data recorded so far.
GET	/v2/samples/{sample_id}	Shows details for a sample, by sample ID.

20.3.1. List samples

Method	URI	Description
GET	/v2/samples{?q, limit}	Lists all known samples, based on the data recorded so far.

Normal response codes: 200

20.3.1.1. Request

This table shows the query parameters for the list samples request:

Name	Type	Description
q	List (Optional)	Filters the response by one or more arguments. For example: ?q.field=Foo&q.value=my_text.
limit	Int (Optional)	Requests a page size of items. Returns a number of items up to a limit value. Use the limit parameter to make an initial limited request and use the ID of the last-seen item from the response as the marker parameter value in a subsequent limited request.

This operation does not accept a request body.

20.3.1.2. Response

Example 20.19. List samples: JSON response

```
[
  {
    "id": "9b23b398-6139-11e5-97e9-bc764e045bf6",
    "metadata": {
      "name1": "value1",
      "name2": "value2"
    },
    "meter": "instance",
    "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
    "recorded_at": "2015-09-22T14:52:54.850725",
    "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
    "source": "openstack",
    "timestamp": "2015-09-22T14:52:54.850718",
    "type": "gauge",
    "unit": "instance",
    "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff",
    "volume": 1
  }
]
```

Example 20.20. List samples: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
  <value>
    <id>9b23b398-6139-11e5-97e9-bc764e045bf6</id>
    <meter>instance</meter>
    <type>gauge</type>
    <unit>instance</unit>
```

```
<volume>1.0</volume>
<user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
<project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
<resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
<source>openstack</source>
<timestamp>2015-09-22T14:52:54.850718</timestamp>
<recorded_at>2015-09-22T14:52:54.850725</recorded_at>
<metadata>
  <item>
    <key>name2</key>
    <value>value2</value>
  </item>
  <item>
    <key>name1</key>
    <value>value1</value>
  </item>
</metadata>
</value>
</values>
```

This operation does not return a response body.

20.3.2. Show sample details

Method	URI	Description
GET	/v2/samples/{sample_id}	Shows details for a sample, by sample ID.

Normal response codes: 200

20.3.2.1. Request

This table shows the URI parameters for the show sample details request:

Name	Type	Description
{sample_id}	UUID	The UUID of the sample.

This operation does not accept a request body.

20.3.2.2. Response

Example 20.21. Show sample details: JSON response

```
{
  "id": "9b23b398-6139-11e5-97e9-bc764e045bf6",
  "metadata": {
    "name1": "value1",
    "name2": "value2"
  },
  "meter": "instance",
  "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
  "recorded_at": "2015-09-22T14:52:54.850725",
  "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
  "source": "openstack",
  "timestamp": "2015-09-22T14:52:54.850718",
  "type": "gauge",
  "unit": "instance",
  "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff",
  "volume": 1
}
```

Example 20.22. Show sample details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
  <id>9b23b398-6139-11e5-97e9-bc764e045bf6</id>
  <meter>instance</meter>
  <type>gauge</type>
  <unit>instance</unit>
  <volume>1.0</volume>
  <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
  <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
  <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
  <source>openstack</source>
  <timestamp>2015-09-22T14:52:54.850718</timestamp>
  <recorded_at>2015-09-22T14:52:54.850725</recorded_at>
  <metadata>
    <item>
```

```
        <key>name2</key>
        <value>value2</value>
    </item>
    <item>
        <key>name1</key>
        <value>value1</value>
    </item>
</metadata>
</value>
```

This operation does not return a response body.

20.4. Resources

Lists all and gets information for resources.

Method	URI	Description
GET	/v2/resources{?q,meter_links}	Lists definitions for all resources.
GET	/v2/resources/{resource_id}	Shows details for a resource, by resource ID.

20.4.1. List resources

Method	URI	Description
GET	/v2/resources{?q,meter_links}	Lists definitions for all resources.

Normal response codes: 200

20.4.1.1. Request

This table shows the query parameters for the list resources request:

Name	Type	Description
q	List (Optional)	Filters the response by one or more arguments. For example: ?q.field=Foo&q.value=my_text.
meter_links	Int (Optional)	Set ?meter_links=1 to return a self link and related meter links.

This operation does not accept a request body.

20.4.1.2. Response

Example 20.23. List resources: JSON response

```
[
  {
    "links": [
      {
        "href": "http://localhost:8777/v2/resources/
bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
        "rel": "self"
      },
      {
        "href": "http://localhost:8777/v2/meters/volume?q.field=
resource_id&q.value=bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
        "rel": "volume"
      }
    ],
    "metadata": {
      "name1": "value1",
      "name2": "value2"
    },
    "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
    "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
    "source": "openstack",
    "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff"
  }
]
```

This table shows the body parameters for the list resources response:

Name	Type	Description
links	List	A list that contains a self link and associated meter links.

Name	Type	Description
	<i>(Required)</i>	
metadata	Dict <i>(Required)</i>	A set of one or more arbitrary metadata key and value pairs that are associated with the resource.
project_id	UUID <i>(Required)</i>	The UUID of the owning project or tenant.
resource_id	UUID <i>(Required)</i>	The UUID of the resource.
source	String <i>(Required)</i>	The name of the source from which the resource came.
user_id	UUID <i>(Required)</i>	The UUID of the user who either created or last updated the resource.

Example 20.24. List resources: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
  <value>
    <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
    <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
    <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
    <metadata>
      <item>
        <key>name2</key>
        <value>value2</value>
      </item>
      <item>
        <key>name1</key>
        <value>value1</value>
      </item>
    </metadata>
    <links>
      <item>
        <href>http://localhost:8777/v2/resources/
bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</href>
        <rel>self</rel>
      </item>
      <item>
        <href>http://localhost:8777/v2/meters/volume?q.field=resource_id&
amp;q.value=bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</href>
        <rel>volume</rel>
      </item>
    </links>
    <source>openstack</source>
  </value>
</values>
```

This operation does not return a response body.

20.4.2. Show resource details

Method	URI	Description
GET	/v2/resources/{resource_id}	Shows details for a resource, by resource ID.

Normal response codes: 200

20.4.2.1. Request

This table shows the URI parameters for the show resource details request:

Name	Type	Description
{resource_id}	UUID	The UUID of the resource.

This operation does not accept a request body.

20.4.2.2. Response

Example 20.25. Show resource details: JSON response

```
{
  "links": [
    {
      "href": "http://localhost:8777/v2/resources/
bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
      "rel": "self"
    },
    {
      "href": "http://localhost:8777/v2/meters/volume?q.field=
resource_id&q.value=bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
      "rel": "volume"
    }
  ],
  "metadata": {
    "name1": "value1",
    "name2": "value2"
  },
  "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
  "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
  "source": "openstack",
  "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff"
}
```

Example 20.26. Show resource details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
  <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
  <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
  <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
  <metadata>
    <item>
      <key>name2</key>
      <value>value2</value>
    </item>
  </metadata>
</value>
```



```
<item>
  <key>name1</key>
  <value>value1</value>
</item>
</metadata>
<links>
  <item>
    <href>http://localhost:8777/v2/resources/
bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</href>
    <rel>self</rel>
  </item>
  <item>
    <href>http://localhost:8777/v2/meters/volume?q.field=resource_id&
amp;q.value=bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</href>
    <rel>volume</rel>
  </item>
</links>
<source>openstack</source>
</value>
```

This operation does not return a response body.

20.5. Capabilities

Gets information for API and storage capabilities.

The Telemetry service enables you to store samples, events, and alarm definitions in supported database back ends. The capabilities resource enables you to list the capabilities that a database supports.

The capabilities resource returns a flattened dictionary of capability properties, each with an associated boolean value. A value of `true` indicates that the corresponding capability is available in the back end.

You can optionally configure separate database back ends for samples, events, and alarms definitions. The capabilities response shows a value of `true` to indicate that the definitions database for samples, events, or alarms is ready to use in a production environment.

Method	URI	Description
GET	/v2/capabilities	A representation of the API and storage capabilities. Usually, the storage driver imposes constraints.

20.5.1. List capabilities

Method	URI	Description
GET	/v2/capabilities	A representation of the API and storage capabilities. Usually, the storage driver imposes constraints.

Normal response codes: 200

20.5.1.1. Response

Example 20.27. List capabilities: JSON response

```
{
  "alarm_storage": {
    "storage:production_ready": true
  },
  "api": {
    "alarms:history:query:complex": true,
    "alarms:history:query:simple": true,
    "alarms:query:complex": true,
    "alarms:query:simple": true,
    "events:query:simple": true,
    "meters:query:complex": false,
    "meters:query:metadata": true,
    "meters:query:simple": true,
    "resources:query:complex": false,
    "resources:query:metadata": true,
    "resources:query:simple": true,
    "samples:query:complex": true,
    "samples:query:metadata": true,
    "samples:query:simple": true,
    "statistics:aggregation:selectable:avg": true,
    "statistics:aggregation:selectable:cardinality": true,
    "statistics:aggregation:selectable:count": true,
    "statistics:aggregation:selectable:max": true,
    "statistics:aggregation:selectable:min": true,
    "statistics:aggregation:selectable:quartile": false,
    "statistics:aggregation:selectable:stddev": true,
    "statistics:aggregation:selectable:sum": true,
    "statistics:aggregation:standard": true,
    "statistics:groupby": true,
    "statistics:query:complex": false,
    "statistics:query:metadata": true,
    "statistics:query:simple": true
  },
  "event_storage": {
    "storage:production_ready": true
  },
  "storage": {
    "storage:production_ready": true
  }
}
```

This table shows the body parameters for the list capabilities response:

Name	Type	Description
api	Dict	A set of key and value pairs that contain the API capabilities for the configured storage driver.

Name	Type	Description
	(Required)	
alarms:history:query:complex	Boolean (Required)	If true, the complex query capability for alarm history is available for the configured database back end.
alarms:history:query:simple	Boolean (Required)	If true, the simple query capability for alarm history is available for the configured database back end.
alarms:query:complex	Boolean (Required)	If true, the complex query capability for alarm definitions is available for the configured database back end.
alarms:query:simple	Boolean (Required)	If true, the simple query capability for alarm definitions is available for the configured database back end.
events:query:simple	Boolean (Required)	If true, the simple query capability for events is available for the configured database back end.
meters:query:complex	Boolean (Required)	If true, the complex query capability for meters is available for the configured database back end.
meters:query:metadata	Boolean (Required)	If true, the simple query capability for the metadata of meters is available for the configured database back end.
meters:query:simple	Boolean (Required)	If true, the simple query capability for meters is available for the configured database back end.
resources:query:complex	Boolean (Required)	If true, the complex query capability for resources is available for the configured database back end.
resources:query:metadata	Boolean (Required)	If true, the simple query capability for the metadata of resources is available for the configured database back end.
resources:query:simple	Boolean (Required)	If true, the simple query capability for resources is available for the configured database back end.
samples:query:complex	Boolean (Required)	If true, the complex query capability for samples is available for the configured database back end.
samples:query:metadata	Boolean (Required)	If true, the simple query capability for the metadata of samples is available for the configured database back end.
samples:query:simple	Boolean (Required)	If true, the simple query capability for samples is available for the configured database back end.
statistics:aggregation:select:avg	Boolean (Required)	If true, the avg capability is available for the configured database back end. Use the avg capability to get average values for samples.
statistics:aggregation:select:cardinality	Boolean (Required)	If true, the cardinality capability is available for the configured database back end. Use the cardinality capability to get cardinality for samples.
statistics:aggregation:select:count	Boolean (Required)	If true, the count capability is available for the configured database back end. Use the count capability to calculate the number of samples for a query.
statistics:aggregation:select:max	Boolean (Required)	If true, the max capability is available for the configured database back end. Use the max capability to calculate the maximum value for a query.
statistics:aggregation:select:min	Boolean (Required)	If true, the min capability is available for the configured database back end. Use the min capability to calculate the minimum value for a query.

Name	Type	Description
statistics:aggregation:select:quartile	Boolean (Required)	If true, the quartile capability is available for the configured database back end. Use the quartile capability to calculate the quartile of sample volumes for a query.
statistics:aggregation:select:stddev	Boolean (Required)	If true, the stddev capability is available for the configured database back end. Use the stddev capability to calculate the standard deviation of sample volumes for a query.
statistics:aggregation:select:sum	Boolean (Required)	If true, the sum capability is available for the configured database back end. Use the sum capability to calculate the sum of sample volumes for a query.
statistics:aggregation:standard	Boolean (Required)	If true, the standard set of aggregation capability is available for the configured database back end.
statistics:groupby	Boolean (Required)	If true, the groupby capability is available for calculating statistics for the configured database back end.
statistics:query:complex	Boolean (Required)	If true, the complex query capability for statistics is available for the configured database back end.
statistics:query:metadata	Boolean (Required)	If true, the simple query capability for the sample metadata that is used to calculate statistics is available for the configured database back end.
statistics:query:simple	Boolean (Required)	If true, the simple query capability for statistics is available for the configured database back end.
alarm_storage	Dict (Required)	Defines the capabilities for the storage that stores persisting alarm definitions. A value of true indicates that the capability is available.
event_storage	Dict (Required)	If true, the capabilities for the storage that stores persisting events is available.
storage	Dict (Required)	If true, the capabilities for the storage that stores persisting samples is available.
storage:production_ready	Boolean (Required)	If true, the database back end is ready to use in a production environment.

Example 20.28. List capabilities: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
  <api>
    <item>
      <key>statistics:query:complex</key>
      <value>>false</value>
    </item>
    <item>
      <key>alarms:history:query:simple</key>
      <value>>true</value>
    </item>
    <item>
      <key>meters:query:metadata</key>
      <value>>true</value>
    </item>
    <item>
      <key>alarms:query:simple</key>
      <value>>true</value>
    </item>
  </api>
</value>
```

```
<key>resources:query:simple</key>
  <value>true</value>
</item>
<item>
  <key>statistics:aggregation:selectable:quartile</key>
  <value>false</value>
</item>
<item>
  <key>statistics:query:simple</key>
  <value>true</value>
</item>
<item>
  <key>statistics:aggregation:selectable:count</key>
  <value>true</value>
</item>
<item>
  <key>statistics:aggregation:selectable:min</key>
  <value>true</value>
</item>
<item>
  <key>statistics:aggregation:selectable:sum</key>
  <value>true</value>
</item>
<item>
  <key>alarms:query:complex</key>
  <value>true</value>
</item>
<item>
  <key>meters:query:complex</key>
  <value>false</value>
</item>
<item>
  <key>statistics:groupby</key>
  <value>true</value>
</item>
<item>
  <key>alarms:history:query:complex</key>
  <value>true</value>
</item>
<item>
  <key>meters:query:simple</key>
  <value>true</value>
</item>
<item>
  <key>samples:query:metadata</key>
  <value>true</value>
</item>
<item>
  <key>statistics:query:metadata</key>
  <value>true</value>
</item>
<item>
  <key>samples:query:simple</key>
  <value>true</value>
</item>
<item>
  <key>resources:query:metadata</key>
  <value>true</value>
</item>
<item>
```

```

        <key>statistics:aggregation:selectable:max</key>
        <value>true</value>
    </item>
    <item>
        <key>samples:query:complex</key>
        <value>true</value>
    </item>
    <item>
        <key>statistics:aggregation:standard</key>
        <value>true</value>
    </item>
    <item>
        <key>events:query:simple</key>
        <value>true</value>
    </item>
    <item>
        <key>statistics:aggregation:selectable:stddev</key>
        <value>true</value>
    </item>
    <item>
        <key>statistics:aggregation:selectable:avg</key>
        <value>true</value>
    </item>
    <item>
        <key>statistics:aggregation:selectable:cardinality</key>
        <value>true</value>
    </item>
    <item>
        <key>resources:query:complex</key>
        <value>false</value>
    </item>
</api>
<storage>
    <item>
        <key>storage:production_ready</key>
        <value>true</value>
    </item>
</storage>
<alarm_storage>
    <item>
        <key>storage:production_ready</key>
        <value>true</value>
    </item>
</alarm_storage>
<event_storage>
    <item>
        <key>storage:production_ready</key>
        <value>true</value>
    </item>
</event_storage>
</value>

```

This operation does not return a response body.

20.6. Events

Lists all events and shows details for an event.

Method	URI	Description
GET	/v2/events{?q, limit}	Lists all events.
GET	/v2/events/{message_id}	Shows details for an event.

20.6.1. List events

Method	URI	Description
GET	/v2/events{?q, limit}	Lists all events.

Normal response codes: 200

20.6.1.1. Request

This table shows the query parameters for the list events request:

Name	Type	Description
q	List (Optional)	Filters the response by one or more event arguments. For example: <code>GET /v2/events?q.field=Foo&q.value=my_text</code>
limit	Int (Optional)	Limits the maximum number of samples that the response returns. For example: <code>GET /v2/events?limit=1000</code>

This operation does not accept a request body.

20.6.1.2. Response

Example 20.29. List events: JSON response

```
[
  {
    "raw": {},
    "traits": [
      {
        "type": "string",
        "name": "action",
        "value": "read"
      },
      {
        "type": "string",
        "name": "eventTime",
        "value": "2015-10-28T20:26:58.545477+0000"
      }
    ],
    "generated": "2015-10-28T20:26:58.546933",
    "message_id": "bae43de6-e9fa-44ad-8c15-40a852584444",
    "event_type": "http.request"
  }
]
```


20.6.2. Show event details

Method	URI	Description
GET	/v2/events/{message_id}	Shows details for an event.

Normal response codes: 200

20.6.2.1. Request

This table shows the URI parameters for the show event details request:

Name	Type	Description
{message_id}	UUID	The UUID of the message.

20.6.2.2. Response

Example 20.30. Show event details: JSON response

```
{
  "raw": {},
  "traits": [
    {
      "type": "string",
      "name": "action",
      "value": "read"
    },
    {
      "type": "string",
      "name": "eventTime",
      "value": "2015-10-28T20:26:58.545477+0000"
    }
  ],
  "generated": "2015-10-28T20:26:58.546933",
  "message_id": "bae43de6-e9fa-44ad-8c15-40a852584444",
  "event_type": "http.request"
}
```