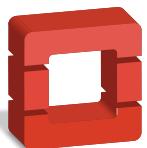


OpenStack

API Complete Reference

(November 17, 2015)



openstack™

OpenStack API Complete Reference

Copyright © 2010-2015 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	6
1.2. API extensions (extensions)	7
1.2.1. List API extensions	8
1.3. Limits (limits)	15
1.3.1. Show absolute limits	16
1.4. Volumes (volumes)	17
1.4.1. Create volume	19
1.4.2. List volumes	23
1.4.3. List volumes with details	25
1.4.4. Show volume details	28
1.4.5. Update volume	30
1.4.6. Delete volume	32
1.5. Volume type access (volumes)	32
1.5.1. Create private volume	33
1.6. Volume actions (volumes, action)	35
1.6.1. Extend volume size	37
1.6.2. Reset volume statuses	38
1.6.3. Set image metadata for volume	39
1.6.4. Remove image metadata from volume	40
1.6.5. Attach volume to server	41
1.6.6. Unmanage volume	42
1.6.7. Force detach volume	43
1.6.8. Promote replicated volume	44
1.6.9. Reenable volume replication	45
1.7. Backups (backups)	45
1.7.1. Create backup	47
1.7.2. List backups	48
1.7.3. List backups with details	50
1.7.4. Show backup details	52
1.7.5. Delete backup	53
1.7.6. Restore backup	54
1.8. Backup actions (backups, action)	54
1.8.1. Force-delete backup	55
1.9. Capabilities for storage back ends (capabilities)	55
1.9.1. Show back-end capabilities	56
1.10. Quota sets extension (os-quota-sets)	57
1.10.1. Show quotas	58
1.10.2. Update quotas	60
1.10.3. Delete quotas	62
1.10.4. Get default quotas	63
1.10.5. Show quotas for user	64
1.10.6. Update quotas for user	66
1.10.7. Delete quotas for user	68
1.10.8. Show quota details for user	69
1.11. Quality of service (QoS) specifications (qos-specs)	69

1.11.1. Create QoS specification	71
1.11.2. List QoS specs	73
1.11.3. Show QoS specification details	75
1.11.4. Set or unset keys in QoS specification	77
1.11.5. Delete QoS specification	78
1.11.6. Associate QoS specification with volume type	79
1.11.7. Disassociate QoS specification from volume type	80
1.11.8. Disassociate QoS specification from all associations	81
1.11.9. Get all associations for QoS specification	82
1.12. Volume types (types)	82
1.12.1. List volume types	84
1.12.2. Create volume type	85
1.12.3. Update volume type	87
1.12.4. Update extra specs for a volume type	89
1.12.5. Show volume type details	91
1.12.6. Delete volume type	92
1.13. Volume snapshots (snapshots)	92
1.13.1. Create snapshot	94
1.13.2. List snapshots	96
1.13.3. List snapshots with details	98
1.13.4. Show snapshot details	100
1.13.5. Update snapshot	101
1.13.6. Delete snapshot	103
1.13.7. Show snapshot metadata	104
1.13.8. Update snapshot metadata	106
1.14. Volume manage extension (os-volume-manage)	107
1.14.1. Manage existing volume	108
1.15. Volume image metadata extension (os-vol-image-meta)	109
1.15.1. Show image metadata for volume	110
1.16. Back-end storage pools	111
1.16.1. List back-end storage pools	112
1.17. Volume transfer	113
1.17.1. Create volume transfer	114
1.17.2. List volume transfers	115
1.17.3. List volume transfers, with details	116
1.17.4. Show volume transfer details	118
1.17.5. Delete volume transfer	119
1.17.6. Accept volume transfer	120
1.18. Consistency groups	120
1.18.1. List consistency groups	122
1.18.2. Create consistency group	123
1.18.3. List consistency groups with details	124
1.18.4. Create consistency group from source	125
1.18.5. Show consistency group details	126
1.18.6. Delete consistency group	127
1.18.7. Update consistency group	128
1.19. Consistency group snapshots	128
1.19.1. List consistency group snapshots	130
1.19.2. Create consistency group snapshot	131
1.19.3. List consistency group snapshots with details	132
1.19.4. Show consistency group snapshot details	133

1.19.5. Delete consistency group snapshot	134
2. Block Storage API v1 (DEPRECATED)	135
2.1. API versions	136
2.1.1. List API versions	137
2.1.2. Show API v1 details	138
2.2. Volumes	139
2.2.1. Create volume	140
2.2.2. List volumes	142
2.2.3. List volumes, with details	144
2.2.4. Show volume details	146
2.2.5. Delete volume	148
2.3. Volume types	148
2.3.1. List volume types	149
2.3.2. Create volume type	150
2.3.3. Show volume type details	152
2.3.4. Delete volume type	153
2.4. Snapshots	153
2.4.1. Create snapshot	154
2.4.2. List snapshots	156
2.4.3. List snapshots with details	158
2.4.4. Show snapshot details	160
2.4.5. Delete snapshot	161
2.4.6. Show snapshot metadata	162
2.4.7. Update snapshot metadata	164
2.5. Quota sets extension (os-quota-sets)	164
2.5.1. Show quotas	166
2.5.2. Update quotas	169
2.5.3. Delete quotas	172
2.5.4. Get default quotas	173
2.5.5. Show quotas for user	175
2.5.6. Update quotas for user	177
2.5.7. Delete quotas for user	180
2.5.8. Show quota details for user	181
3. Compute API (CURRENT)	183
3.1. API versions	192
3.1.1. List API versions	194
3.2. Servers (servers)	194
3.2.1. List servers	197
3.2.2. Create server	200
3.2.3. List details for servers	203
3.2.4. Get server details	206
3.2.5. Update server	209
3.2.6. Delete server	213
3.3. Servers multiple create (servers)	214
3.3.1. Create multiple servers	215
3.3.2. Create multiple servers with reservation ID	217
3.4. Servers actions (servers, action)	218
3.4.1. Add fixed IP to server (addFixedIp action)	220
3.4.2. Attach volume (attach action)	221
3.4.3. Confirm resized server (confirmResize action)	222
3.4.4. Create image (createImage action)	223

3.4.5. Evacuate server (evacuate action)	225
3.4.6. Force-delete server (forceDelete action)	226
3.4.7. Get console output for a server (os-getConsoleOutput action)	227
3.4.8. Get RDP console for a server (os-getRDPConsole action)	228
3.4.9. Get serial console for a server (os-getSerialConsole action)	229
3.4.10. Get SPICE console for a server (os-getSPICEConsole action)	230
3.4.11. Get VNC console for a server (os-getVNCConsole action)	231
3.4.12. Reboot server (reboot action)	232
3.4.13. Rebuild server (rebuild action)	233
3.4.14. Remove fixed IP from server (removeFixedIp action)	236
3.4.15. Rescue server (rescue action)	237
3.4.16. Resize server (resize action)	238
3.4.17. Restore soft-deleted instance (restore action)	239
3.4.18. Revert resized server (revertResize action)	240
3.4.19. Shelve server (shelve action)	241
3.4.20. Shelf-offload (remove) server (shelveOffload action)	243
3.4.21. Start server (os-start action)	244
3.4.22. Stop server (os-stop action)	245
3.4.23. Unrescue server (unrescue action)	246
3.4.24. Unshelve (restore) shelved server (unshelve action)	247
3.5. Servers admin actions (servers, action)	247
3.5.1. Change administrative password (changePassword action)	249
3.5.2. Create server back up (createBackup action)	250
3.5.3. Inject network information (injectNetworkInfo action)	251
3.5.4. Lock server (lock action)	252
3.5.5. Migrate server (migrate action)	253
3.5.6. Live-migrate server (os-migrateLive action)	254
3.5.7. Pause server (pause action)	255
3.5.8. Reset networking on a server (resetNetwork action)	256
3.5.9. Reset server state (os-resetState action)	257
3.5.10. Resume suspended server (resume action)	258
3.5.11. Suspend server (suspend action)	259
3.5.12. Unlock server (unlock action)	260
3.5.13. Unpause server (unpause action)	261
3.6. Servers diagnostics (servers, diagnostics)	261
3.6.1. Get server diagnostics	262
3.7. Servers IPs (servers, ips)	262
3.7.1. List IPs	264
3.7.2. Show IP details	265
3.8. Server metadata (servers, metadata)	265
3.8.1. List all metadata	267
3.8.2. Create or replace metadata items	268
3.8.3. Update metadata items	269
3.8.4. Show metadata item details	270
3.8.5. Create or update metadata item	271
3.8.6. Delete metadata item	272
3.9. Servers action (servers, os-instance-actions)	272
3.9.1. List actions for server	273
3.9.2. Show server action details	274
3.10. Servers password (servers, os-server-password)	274
3.10.1. Get server password	276

3.11. Servers virtual interfaces (servers, os-virtual-interfaces)	276
3.11.1. List virtual interfaces	277
3.12. Flavors with extended attributes (flavors)	277
3.12.1. List flavors	278
3.12.2. Show flavor details	280
3.12.3. List flavors with details	281
3.13. Flavors access (flavors, os-flavor-access, action)	283
3.13.1. Create flavor with access list	284
3.13.2. List flavors with access list	286
3.13.3. Show flavor details with access list	287
3.13.4. Add flavor access to tenant	288
3.13.5. Remove flavor access from tenant	289
3.13.6. List flavors with details and access list	290
3.14. Flavors extra-specs (flavors, os-flavor-extra-specs)	292
3.14.1. Create or update flavor extra specs	293
3.14.2. Show flavor extra specs	294
3.14.3. Show flavor extra specs	295
3.14.4. Update flavor extra spec	296
3.15. Flavors manage (flavors, os-flavor-manage)	296
3.15.1. Create flavor	297
3.15.2. Delete flavor	298
3.16. Keypairs (keypairs)	298
3.16.1. List keypairs	299
3.16.2. Create or import keypair	300
3.16.3. Delete keypair	301
3.16.4. Show keypair information	302
3.17. Limits (limits)	302
3.17.1. Show global and rate limits	303
3.18. Extensions (extensions)	303
3.18.1. List extensions	305
3.18.2. Get extension	333
3.19. Images	333
3.19.1. List images	335
3.19.2. List images details	342
3.19.3. Get image details	352
3.19.4. Delete image	354
3.20. Image metadata	354
3.20.1. Show image metadata	355
3.20.2. Create or replace image metadata	356
3.20.3. Update image metadata items	357
3.20.4. Show image metadata item details	358
3.20.5. Create or update image metadata item	359
3.20.6. Delete image metadata item	360
3.21. Guest agents (os-agents)	360
3.21.1. Create agent build	361
3.21.2. List agent builds	362
3.21.3. Delete agent build	363
3.21.4. Update agent build	364
3.22. Host aggregates (os-aggregates, action)	364
3.22.1. Create aggregate	366
3.22.2. List aggregates	367

3.22.3. Show aggregate details	368
3.22.4. Update aggregate	369
3.22.5. Add host	370
3.22.6. Create or update aggregate metadata	371
3.22.7. Remove host	372
3.23. Assisted volume snapshots (os-assisted-volume-snapshots)	372
3.23.1. Create assisted volume snapshots	374
3.23.2. Delete assisted volume snapshot	375
3.24. Attach Interfaces (os-attach-interfaces)	375
3.24.1. Create interface	376
3.24.2. List interfaces	377
3.24.3. Show port interface details	378
3.24.4. Detach interface	379
3.25. Availability zones (os-availability-zone)	379
3.25.1. Get availability zone information	380
3.25.2. Get detailed availability zone information	381
3.26. Bare metal nodes (os-baremetal-nodes)	382
3.26.1. Add bare metal node	383
3.26.2. List bare metal nodes	385
3.26.3. Add interface to bare metal node	386
3.26.4. Delete interface from bare metal node	387
3.26.5. Show bare metal node details	388
3.26.6. Delete bare metal node	390
3.27. Servers with block device mapping format (servers, os-block-device-mapping)	390
3.27.1. Create server with block device mapping	391
3.28. Cells (os-cells, capacities)	392
3.28.1. List cells	393
3.28.2. List cells with details	395
3.28.3. Show cell data	396
3.28.4. Show cell capacities	397
3.29. Root certificates (os-certificates)	398
3.29.1. Create certificate	399
3.29.2. Show certificate details	401
3.30. Cloudpipe (os-cloudpipe)	401
3.30.1. List cloudpipes	403
3.30.2. Create cloudpipe	404
3.30.3. Update cloudpipe	405
3.31. Server consoles (servers, os-consoles, os-console-auth-token)	405
3.31.1. Create console for server	406
3.31.2. Lists consoles for server	407
3.31.3. Show console details	408
3.31.4. Delete console	409
3.31.5. Show console authentication token	410
3.32. Fixed IPs (os-fixed-ips)	410
3.32.1. Show fixed IP details	411
3.32.2. Reserve or release a fixed IP	412
3.33. Floating IP DNS records (os-floating-ip-dns)	412
3.33.1. List DNS domains	413
3.33.2. Create or update DNS domain	414
3.33.3. Delete DNS domain	415

3.33.4. Create or update DNS entry	416
3.33.5. Find unique DNS entry	417
3.33.6. Delete DNS entry	418
3.33.7. List DNS entries	419
3.34. Floating IP pools (os-floating-ip-pools)	419
3.34.1. List floating IP pools	420
3.35. Floating IPs (os-floating-ips)	420
3.35.1. List floating IPs	421
3.35.2. Allocate floating IP	422
3.35.3. Deallocate floating IP	423
3.36. Floating IPs bulk (os-floating-ips-bulk)	423
3.36.1. List floating IPs	424
3.36.2. Create floating IPs	425
3.36.3. Bulk-delete floating IPs	426
3.36.4. List floating IPs by host	427
3.37. Ping instances (os-fping)	427
3.37.1. Ping instances	428
3.37.2. Ping an instance	430
3.38. Hosts (os-hosts)	430
3.38.1. List hosts	431
3.38.2. Enable host	433
3.38.3. Show host details	434
3.38.4. Reboot host	435
3.38.5. Shut down host	436
3.38.6. Start host	437
3.39. Hypervisors (os-hypervisors)	437
3.39.1. List hypervisors	438
3.39.2. Show hypervisor statistics	439
3.39.3. Show hypervisor details	440
3.39.4. Show hypervisor uptime	442
3.40. Instance usage audit log (os-instance-usage-audit-log)	442
3.40.1. List usage audits for an instance	443
3.40.2. List usage audits before specified time	444
3.41. Migrations (os-migrations)	445
3.41.1. List migrations	446
3.42. Networks (os-networks)	447
3.42.1. Create network	448
3.42.2. List networks	450
3.42.3. Add network	452
3.42.4. Show network details	453
3.42.5. Delete network	455
3.42.6. Associate host	456
3.42.7. Disassociate host	457
3.42.8. Disassociate network	458
3.42.9. Disassociate project	459
3.43. Quota class (os-quota-class-sets)	459
3.43.1. Show quota	460
3.43.2. Update quota	461
3.44. Quota sets (os-quota-sets)	462
3.44.1. Delete quota for tenant	463
3.44.2. Force-update quota	464

3.44.3. Update quota	465
3.44.4. Show default quotas	466
3.44.5. Show quotas (detailed)	467
3.44.6. Update quota for user	468
3.44.7. Delete quota for user	469
3.45. Security groups (os-security-groups)	469
3.45.1. List security groups	470
3.45.2. Create security group	471
3.45.3. Show security group details	472
3.45.4. Update security group	473
3.45.5. Delete security group	474
3.45.6. List security groups by server	475
3.46. Rules for default security group (os-security-group-default-rules)	475
3.46.1. List default security group rules	476
3.46.2. Create default security group rule	477
3.46.3. Show default security group rule details	478
3.46.4. Delete default security group rule	479
3.47. Rules for security group (os-security-group-rules)	479
3.47.1. Create security group rule	480
3.47.2. Delete security group rule	481
3.48. Execute external events (os-server-external-events)	481
3.48.1. Create events	482
3.49. Server groups (os-server-groups)	482
3.49.1. List server groups	484
3.49.2. Create server group	485
3.49.3. Show server group details	486
3.49.4. Delete server group	487
3.50. Manage services (os-services)	487
3.50.1. List services	488
3.50.2. List disabled services	490
3.50.3. Enable scheduling for a service	492
3.50.4. Disable scheduling for a service	493
3.50.5. Log disabled service information	494
3.50.6. Force down service	495
3.50.7. Delete service	496
3.51. Usage reports (os-simple-tenant-usage)	496
3.51.1. List tenant usage for all tenants	497
3.51.2. Show usage details for tenant	498
3.52. Project networks (os-tenant-networks)	498
3.52.1. Create project network	500
3.52.2. List project networks	501
3.52.3. Show project network details	502
3.52.4. Delete project network	503
3.53. Volume extension (os-volumes, os-snapshots)	503
3.53.1. List volumes	504
3.53.2. Create volume	505
3.53.3. List volumes with details	506
3.53.4. Show volume details	507
3.53.5. Delete volume	508
3.53.6. List volume types	509
3.53.7. Show volume type details	510

3.53.8. Create snapshot	511
3.53.9. List snapshots	512
3.53.10. List snapshots with details	513
3.53.11. Show snapshot details	515
3.53.12. Delete snapshot	516
4. Database Service API v1.0 (CURRENT)	517
4.1. API versions	518
4.1.1. List versions	519
4.1.2. Show version details	520
4.2. Database instances (instances)	520
4.2.1. Create database instance	522
4.2.2. List database instances	525
4.2.3. Show database instance details	527
4.2.4. Delete database instance	529
4.2.5. Attach configuration group	530
4.2.6. Detach configuration group	531
4.2.7. List configuration defaults	532
4.2.8. Enable root user	534
4.2.9. Show root-enabled status for database instance	536
4.3. Database instance actions (action)	536
4.3.1. Resize instance	538
4.3.2. Resize instance volume	539
4.3.3. Restart instance	541
4.4. Databases (databases)	542
4.4.1. Create database	543
4.4.2. List instance databases	545
4.4.3. Delete database	547
4.5. Users (users)	547
4.5.1. Create user	549
4.5.2. List database instance users	552
4.5.3. Delete user	554
4.6. Flavors (flavors)	555
4.6.1. List flavors	556
4.6.2. Show flavor details	558
4.7. Datastores (datastores)	559
4.7.1. List datastore versions	560
4.7.2. List configuration parameters	562
4.7.3. Show configuration parameter details	568
4.8. Configuration groups (configurations)	569
4.8.1. Create configuration group	570
4.8.2. List configuration groups	572
5. Data Processing API v1.1 (CURRENT)	574
5.1. Plugins	576
5.1.1. List plugins	577
5.1.2. Show plugin details	579
5.1.3. Show plugin version details	580
5.2. Image registry	582
5.2.1. List images	583
5.2.2. Show image details	585
5.2.3. Register image	586
5.2.4. Unregister image	587

5.2.5. Add tags to image	588
5.2.6. Remove tags from image	590
5.3. Node group templates	591
5.3.1. List node group templates	592
5.3.2. Create node group template	594
5.3.3. Show node group template details	596
5.3.4. Delete node group template	598
5.3.5. Update node group template	599
5.4. Cluster templates	600
5.4.1. List available cluster templates	601
5.4.2. Create cluster templates	604
5.4.3. Show cluster template	607
5.4.4. Update cluster templates	610
5.4.5. Delete cluster template	612
5.5. Clusters	612
5.5.1. List available clusters	613
5.5.2. Create cluster	620
5.5.3. Create multiple clusters	623
5.5.4. Show details of a cluster	624
5.5.5. Delete a cluster	627
5.5.6. Scale cluster	628
5.5.7. Update cluster	636
5.6. Event log	638
5.6.1. Show progress	639
5.7. Data sources	640
5.7.1. List data sources	642
5.7.2. Create data source	643
5.7.3. Show data source details	645
5.7.4. Delete data source	646
5.7.5. Update data source	647
5.8. Job binary internals	647
5.8.1. List job binary internals	649
5.8.2. Create job binary internal	650
5.8.3. Show job binary internal details	651
5.8.4. Delete job binary internal	652
5.8.5. Update job binary internal	653
5.8.6. Get job binary internal data	654
5.9. Job binaries	654
5.9.1. List job binaries	656
5.9.2. Create job binary	658
5.9.3. Show job binary details	659
5.9.4. Delete job binary	660
5.9.5. Update job binary	661
5.9.6. Get job binary data	662
5.10. Jobs	662
5.10.1. List jobs	664
5.10.2. Create job	666
5.10.3. Show job details	668
5.10.4. Remove job	669
5.10.5. Update job object	670
5.10.6. Run job	671

5.11. Job executions	672
5.11.1. List job executions	673
5.11.2. Show job execution details	676
5.11.3. Delete job execution	679
5.11.4. Update job execution	680
5.11.5. Refresh job execution status	683
5.11.6. Cancel job execution	686
5.12. Job types	688
5.12.1. List job types	689
6. Identity API v3 (CURRENT)	694
6.1. API versions	696
6.1.1. List versions	697
6.1.2. Show API version details	699
6.2. Tokens	699
6.2.1. Authenticate	700
6.2.2. Validate token	709
6.2.3. Check token	711
6.2.4. Revoke token	712
6.3. Service catalog	712
6.3.1. Create service	713
6.3.2. List services	714
6.3.3. Show service details	716
6.3.4. Update service	717
6.3.5. Delete service	718
6.4. Endpoints	718
6.4.1. Create endpoint	719
6.4.2. List endpoints	720
6.4.3. Update endpoint	722
6.4.4. Delete endpoint	724
6.5. Domains	724
6.5.1. Create domain	726
6.5.2. List domains	727
6.5.3. Show domain details	729
6.5.4. Update domain	730
6.5.5. Delete domain	731
6.5.6. List roles for domain user	732
6.5.7. Grant role to domain user	734
6.5.8. Check role for domain user	735
6.5.9. Revoke role from domain user	736
6.5.10. List roles for domain group	737
6.5.11. Grant role to domain group	739
6.5.12. Check role for domain group	740
6.5.13. Revoke role from domain group	741
6.6. Projects	741
6.6.1. Create project	743
6.6.2. List projects	744
6.6.3. Show project details	746
6.6.4. Update project	747
6.6.5. Delete project	749
6.6.6. List roles for project user	750
6.6.7. Grant role to project user	752

6.6.8. Check role for project user	753
6.6.9. Revoke role from project user	754
6.6.10. List roles for project group	755
6.6.11. Grant role to project group	757
6.6.12. Check role for project group	758
6.6.13. Revoke role from project group	759
6.7. Users	759
6.7.1. Create user	760
6.7.2. List users	761
6.7.3. Show user details	763
6.7.4. Update user	764
6.7.5. Delete user	766
6.7.6. List groups for user	767
6.7.7. List projects for user	769
6.8. Groups	770
6.8.1. Create group	771
6.8.2. List groups	772
6.8.3. Show group details	774
6.8.4. Update group	775
6.8.5. Delete group	776
6.8.6. List users in group	777
6.8.7. Assign user to group	779
6.8.8. Remove user from group	780
6.8.9. Check user membership in group	781
6.9. Credentials	781
6.9.1. Create credential	782
6.9.2. List credentials	783
6.9.3. Show credential details	785
6.9.4. Update credential	786
6.9.5. Delete credential	787
6.10. Roles	787
6.10.1. Create role	788
6.10.2. List roles	789
6.10.3. List role assignments	791
6.10.4. Show role details	794
6.10.5. Update role	795
6.10.6. Delete role	796
6.11. Policies	796
6.11.1. Create policy	798
6.11.2. List policies	799
6.11.3. Show policy details	801
6.11.4. Update policy	802
6.11.5. Delete policy	804
6.12. Regions	804
6.12.1. List regions	805
6.12.2. Create region	806
6.12.3. Create region with user-defined ID	808
6.12.4. Show region details	810
6.12.5. Update region	811
6.12.6. Delete region	812
7. Identity API v3 extensions (CURRENT)	813

7.1. OS-INHERIT extension (OS-INHERIT)	815
7.1.1. List role assignments	817
7.1.2. List roles for user	820
7.1.3. Assign role to user	821
7.1.4. Check role for user	822
7.1.5. Revoke role from user	823
7.1.6. List roles for group	824
7.1.7. Assign role to group	825
7.1.8. Check role for group	826
7.1.9. Revoke role from group	827
7.1.10. List project roles for user on domain	828
7.1.11. Assign role to user owned by domain projects	829
7.1.12. Check project role for user on domain	830
7.1.13. Revoke project role from user on domain	831
7.1.14. List project roles for group on domain	832
7.1.15. Assign role to group on domain projects	833
7.1.16. Check project role for group on domain	834
7.1.17. Revoke project role from group on domain	835
7.2. Key Distribution Server (KDS) extension (OS-KDS)	835
7.2.1. Create key	839
7.2.2. Delete key	840
7.2.3. Generate ticket	841
7.2.4. Get group key	842
7.2.5. Create group	843
7.2.6. Delete group	844
7.3. OAuth extension (OS-OAUTH1)	844
7.3.1. Create consumer	846
7.3.2. List consumers	847
7.3.3. Show consumer details	848
7.3.4. Update consumer	849
7.3.5. Delete consumer	850
7.3.6. Get unauthorized request token	851
7.3.7. Create access token	852
7.3.8. List authorized access tokens	853
7.3.9. Get authorized access token	854
7.3.10. Revoke access token	855
7.3.11. List roles for an access token	856
7.3.12. Show role details for an access token	857
7.3.13. Get an Identity Service token	858
7.4. Trust extension (OS-TRUST)	858
7.4.1. Create trust	860
8. Identity API v2.0 (SUPPORTED)	862
8.1. API versions	862
8.1.1. List versions	863
8.1.2. Show version details	865
8.2. Tokens	865
8.2.1. Authenticate	866
8.2.2. List tenants	875
9. Identity Admin API v2.0 (SUPPORTED)	876
9.1. Versions	876
9.1.1. Get version details	877

9.2. Tokens	878
9.2.1. Authenticate for admin API	879
9.2.2. Validate token	887
9.2.3. Validate token (admin)	889
9.2.4. Delete token	890
9.3. Users	890
9.3.1. Create user	891
9.3.2. List users	892
9.3.3. Update user	896
9.3.4. Delete user	897
9.3.5. Get user information by ID	898
9.3.6. List user global roles	899
9.4. Tenants	899
9.4.1. List tenants	901
9.4.2. Get tenant information by name	903
9.4.3. Get tenant information by ID	904
9.4.4. List roles for user	905
10. Identity API v2.0 extensions (SUPPORTED)	906
10.1. OS-KSADM admin extension	907
10.1.1. List users	909
10.1.2. Create user	910
10.1.3. Update user	911
10.1.4. Delete user	912
10.1.5. Enable user	913
10.1.6. List global roles for user	914
10.1.7. Grant global role to user	916
10.1.8. Delete global role from user	917
10.1.9. Create tenant	918
10.1.10. Update tenant	919
10.1.11. Delete tenant	920
10.1.12. List users on a tenant	921
10.1.13. Grant roles to user on tenant	923
10.1.14. Revoke role from user on tenant	924
10.1.15. Create role	925
10.1.16. Show role information by name	926
10.1.17. Show role details, by ID	927
10.1.18. Delete role	928
10.1.19. List roles	929
10.1.20. List services	930
10.1.21. Create service	932
10.1.22. Show service information by name	933
10.1.23. Shows service information by ID	934
10.1.24. Delete service	935
10.2. OS-KSCATALOG admin extension	935
10.2.1. List endpoints	936
10.2.2. Create endpoint	938
10.2.3. Get endpoint	939
10.2.4. Delete endpoint	940
10.2.5. List endpoint templates	941
10.2.6. Create endpoint template	943
10.2.7. Get endpoint template	945

10.2.8. Update endpoint template	946
10.2.9. Delete endpoint template.	948
10.3. OS-KSEC2 admin extension	948
10.3.1. List credentials	949
10.3.2. Grant credential to user	951
10.3.3. Get user credentials	952
10.3.4. Update user credentials	953
10.3.5. Delete user credentials	954
10.3.6. List credentials by type	955
10.4. OS-KSS3 admin extension	956
10.4.1. List credentials	957
10.4.2. Grant credential to user	959
10.4.3. Get user credentials	960
10.4.4. Update user credentials	961
10.4.5. Revoke user credentials	962
10.4.6. List credentials by type	963
10.5. OS-KVALIDATE admin extension	964
10.5.1. Validate token	965
10.5.2. Check token	967
10.5.3. List endpoints for token	968
11. Image service API v2 (CURRENT)	970
11.1. API versions	972
11.1.1. List API versions	973
11.2. Images	974
11.2.1. Create image	975
11.2.2. List images	977
11.2.3. Show image details	981
11.2.4. Update image	982
11.2.5. Delete image	985
11.2.6. Reactivate image	986
11.2.7. Deactivate image	987
11.3. Image data	988
11.3.1. Upload binary image data	989
11.3.2. Download binary image data	990
11.4. Image tags	990
11.4.1. Add image tag	992
11.4.2. Delete image tag	993
11.5. Members	993
11.5.1. Create image member	994
11.5.2. List image members	996
11.5.3. Show image member details	998
11.5.4. Delete image member	999
11.5.5. Update image member	1000
11.6. Image schemas	1000
11.6.1. Get images schema	1002
11.6.2. Get image schema	1007
11.6.3. Get image members schema	1012
11.6.4. Get image member schema	1014
11.7. Metadata definition resource types (since API v2.0)	1015
11.7.1. List resource types	1016
11.7.2. Create resource type association	1017

11.7.3. List resource type associations	1018
11.7.4. Remove resource type association	1019
11.8. Metadata definition namespaces (since API v2.0)	1019
11.8.1. Create namespace	1020
11.8.2. List namespaces	1022
11.8.3. Get namespaces details	1025
11.8.4. Update namespace	1028
11.8.5. Delete namespace	1029
11.9. Metadata definition properties (since API v2.0)	1029
11.9.1. Create property	1030
11.9.2. List properties	1032
11.9.3. Get property definition	1035
11.9.4. Update property definition	1036
11.9.5. Remove property definition	1037
11.10. Metadata definition objects (since API v2.0)	1037
11.10.1. Create object	1038
11.10.2. List objects	1040
11.10.3. Get object definition	1044
11.10.4. Update object definition	1046
11.10.5. Delete property definition	1048
11.11. Metadata definition tags (since API v2.0)	1048
11.11.1. Create tags	1049
11.11.2. List tags	1050
11.11.3. Delete all tag definitions	1051
11.11.4. Add tag definition	1052
11.11.5. Get tag definition	1053
11.11.6. Update tag definition	1054
11.11.7. Delete tag definition	1055
11.12. Metadata definition schemas	1055
11.12.1. Get metadata definition namespace schema	1056
11.12.2. Get metadata definition namespaces schema	1060
11.12.3. Get metadata definition object schema	1065
11.12.4. Get metadata definition objects schema	1068
11.12.5. Get metadata definition property schema	1072
11.12.6. Get metadata definition properties schema	1075
11.12.7. Get metadata definition tag schema	1078
11.12.8. Get metadata definition tags schema	1079
11.12.9. Get metadata definition namespace resource type association schema	1081
11.12.10. Get metadata definition namespace resource type associations schema	1083
11.13. Tasks (since API v2.2)	1084
11.13.1. Create tasks	1086
11.13.2. List tasks	1087
11.13.3. Show tasks	1088
12. Image service API v1 (SUPPORTED)	1089
12.1. API versions	1089
12.1.1. List API versions	1090
12.2. Images	1091
12.2.1. List images	1092
12.2.2. Create image	1094

12.2.3. List image details	1095
12.2.4. Show image details and image binary	1097
12.2.5. Update image	1098
12.2.6. Delete image	1099
12.3. Members	1099
12.3.1. Replace member	1100
12.3.2. Add member	1101
12.3.3. Remove member	1102
12.4. Shared images	1102
12.4.1. List shared images	1103
13. Networking API v2.0 (CURRENT)	1104
13.1. API versions	1105
13.1.1. List API versions	1106
13.1.2. Show API v2.0 details	1107
13.2. Networks	1110
13.2.1. List networks	1111
13.2.2. Create network	1113
13.2.3. Bulk create networks	1115
13.2.4. Show network details	1117
13.2.5. Update network	1119
13.2.6. Delete network	1121
13.3. Subnets	1121
13.3.1. List subnets	1122
13.3.2. Create subnet	1125
13.3.3. Bulk create subnet	1127
13.3.4. Show subnet details	1130
13.3.5. Update subnet	1132
13.3.6. Delete subnet	1134
13.4. Ports	1134
13.4.1. List ports	1135
13.4.2. Create port	1138
13.4.3. Bulk create ports	1140
13.4.4. Show port details	1143
13.4.5. Update port	1145
13.4.6. Delete port	1148
13.5. Service providers	1148
13.5.1. List service providers	1149
14. Networking API v2.0 extensions (CURRENT)	1150
14.1. Extensions	1154
14.1.1. List extensions	1155
14.1.2. Show extension details	1165
14.2. Quotas extension (quotas)	1166
14.2.1. List quotas	1167
14.2.2. Show quota	1168
14.2.3. Update quota	1169
14.2.4. Reset quota	1171
14.3. Networks provider extended attributes (networks)	1171
14.3.1. List networks	1172
14.3.2. Create network	1173
14.3.3. Show network details	1175
14.3.4. Update network	1177

14.3.5. Delete network	1179
14.4. Networks multiple provider extension (networks)	1179
14.4.1. List networks	1180
14.4.2. Create network with multiple segment mappings	1182
14.4.3. Show details for a network with multiple segments	1184
14.5. VLAN transparency extension (networks)	1184
14.5.1. List networks with VLAN transparency attribute	1186
14.5.2. Create VLAN-transparent network	1187
14.5.3. Show VLAN-transparent network details	1188
14.6. Ports binding extended attributes (ports)	1188
14.6.1. List ports	1189
14.6.2. Create port	1193
14.6.3. Show port details	1195
14.6.4. Update port	1197
14.6.5. Delete port	1199
14.7. Security groups (security-groups)	1199
14.7.1. List security groups	1200
14.7.2. Create security group	1202
14.7.3. Show security group	1204
14.7.4. Update security group	1206
14.7.5. Delete security group	1207
14.8. Security group rules (security-group-rules)	1207
14.8.1. List security group rules	1208
14.8.2. Create security group rule	1210
14.8.3. Show security group rule	1211
14.8.4. Delete security group rule	1212
14.9. Layer-3 networking	1212
14.9.1. List routers	1214
14.9.2. Create router	1216
14.9.3. Show router details	1218
14.9.4. Update router	1219
14.9.5. Delete router	1221
14.9.6. Add interface to router	1222
14.9.7. Remove interface from router	1224
14.9.8. List floating IPs	1226
14.9.9. Create floating IP	1227
14.9.10. Show floating IP details	1229
14.9.11. Update floating IP	1230
14.9.12. Delete floating IP	1232
14.10. Metering labels and rules	1232
14.10.1. List metering labels	1233
14.10.2. Create metering label	1234
14.10.3. Show metering label	1235
14.10.4. Delete metering label	1236
14.10.5. List metering label rules	1237
14.10.6. Create metering label rule	1238
14.10.7. Show metering label rule	1239
14.10.8. Delete metering label rule	1241
14.11. Firewall-as-a-Service (FWaaS) 2.0 (CURRENT)	1241
14.11.1. List firewalls	1243
14.11.2. Create firewall	1244

14.11.3. Show firewall details	1245
14.11.4. Update firewall	1246
14.11.5. Remove firewall	1247
14.12. Load-Balancer-as-a-Service (LBaaS) 1.0 (STABLE)	1247
14.12.1. List VIPs	1249
14.12.2. Create a load balancer VIP	1250
14.12.3. Show VIP details	1251
14.12.4. Update VIP	1252
14.12.5. Delete VIP	1253
14.12.6. List health monitors	1254
14.12.7. Create a load balancer health monitor	1255
14.12.8. Show health monitor details	1256
14.12.9. Update health monitor	1257
14.12.10. Delete health monitor	1258
14.12.11. List pools	1259
14.12.12. Create a load balancer pool	1260
14.12.13. Show pool details	1261
14.12.14. Update pool	1262
14.12.15. Delete pool	1263
14.12.16. Associate health monitor with pool	1264
14.12.17. Disassociate health monitor from pool	1265
14.12.18. List members	1266
14.12.19. Create a load balancer member	1267
14.12.20. Show member details	1268
14.12.21. Update member	1269
14.12.22. Delete member	1270
14.13. Load-Balancer-as-a-Service (LBaaS) 2.0 (EXPERIMENTAL)	1270
14.13.1. List load balancers	1272
14.13.2. Create load balancer	1274
14.13.3. Show load balancer details	1276
14.13.4. Update load balancer	1278
14.13.5. Remove load balancer	1280
14.13.6. List listeners	1281
14.13.7. Create listener	1283
14.13.8. Show listener details	1286
14.13.9. Update listener	1288
14.13.10. Remove listener	1290
14.13.11. List pools	1291
14.13.12. Create pool	1293
14.13.13. Show pool details	1295
14.13.14. Update pool	1297
14.13.15. Remove pool	1299
14.13.16. List pool members	1300
14.13.17. Add member to pool	1302
14.13.18. Show pool member details	1304
14.13.19. Update pool member	1306
14.13.20. Remove member from pool	1307
14.13.21. Create health monitor	1308
14.13.22. Show health monitor details	1310
14.13.23. Update health monitor	1312
14.13.24. Remove health monitor	1314

14.14. Subnet pools extension (subnetpools)	1314
14.14.1. List subnet pools	1315
14.14.2. Create subnet pool	1317
14.14.3. Show subnet pool	1318
14.14.4. Update subnet pool	1319
14.14.5. Delete subnet pool	1321
14.15. Virtual-Private-Network-as-a-Service (VPNaaS) 2.0 (CURRENT)	1321
14.15.1. List VPN services	1323
14.15.2. Create VPN service	1324
14.15.3. Show VPN service details	1325
14.15.4. Update VPN service	1326
14.15.5. Remove VPN service	1327
14.15.6. List IKE policies	1328
14.15.7. Create IKE policy	1329
14.15.8. Show IKE policy details	1330
14.15.9. Update IKE policy	1331
14.15.10. Remove IKE policy	1332
14.15.11. List IPSec policies	1333
14.15.12. Create IPSec policy	1334
14.15.13. Show IPSec policy	1335
14.15.14. Update IPSec policy	1336
14.15.15. Remove IPSec policy	1337
14.15.16. List VPN endpoint groups	1338
14.15.17. Create VPN endpoint group	1339
14.15.18. Show VPN endpoint group	1340
14.15.19. Update VPN endpoint group	1341
14.15.20. Remove VPN endpoint group	1342
14.15.21. List IPSec connections	1343
14.15.22. Create IPSec connection	1344
14.15.23. Show IPSec connection	1346
14.15.24. Update IPSec connection	1347
14.15.25. Remove IPSec connection	1349
14.16. Extra routes	1349
14.16.1. Update router	1350
15. Object Storage API v1 (SUPPORTED)	1352
15.1. Discoverability	1353
15.1.1. List activated capabilities	1354
15.2. Endpoints	1354
15.2.1. List endpoints	1355
15.3. Accounts	1355
15.3.1. Show account details and list containers	1357
15.3.2. Create, update, or delete account metadata	1361
15.3.3. Show account metadata	1364
15.4. Containers	1366
15.4.1. Show container details and list objects	1367
15.4.2. Create container	1371
15.4.3. Delete container	1374
15.4.4. Create, update, or delete container metadata	1376
15.4.5. Show container metadata	1380
15.5. Objects	1382
15.5.1. Get object content and metadata	1383

15.5.2. Create or replace object	1387
15.5.3. Copy object	1391
15.5.4. Delete object	1395
15.5.5. Show object metadata	1397
15.5.6. Create or update object metadata	1400
16. Orchestration API v1 (CURRENT)	1403
16.1. General API information	1405
16.2. API versions	1405
16.2.1. List versions	1406
16.3. Stacks	1406
16.3.1. Create stack	1408
16.3.2. Adopt stack	1410
16.3.3. List stack data	1412
16.3.4. Preview stack	1415
16.3.5. Preview stack update	1419
16.3.6. Find stack	1422
16.3.7. Find stack resources	1423
16.3.8. Show stack details	1424
16.3.9. Update stack	1426
16.3.10. Delete stack	1428
16.3.11. Abandon stack	1429
16.3.12. Snapshot stack	1431
16.3.13. List snapshots	1432
16.3.14. Show snapshot	1433
16.3.15. Delete snapshot	1435
16.3.16. Restore snapshot	1436
16.3.17. List outputs	1437
16.3.18. Show output	1438
16.4. Stack actions	1438
16.4.1. Suspend stack	1439
16.4.2. Resume stack	1440
16.4.3. Cancel stack update	1441
16.4.4. Check stack resources	1442
16.5. Stack resources	1442
16.5.1. List resources	1443
16.5.2. Show resource data	1445
16.5.3. Show resource metadata	1447
16.5.4. Send a signal to a resource	1448
16.6. Stack events	1448
16.6.1. Find stack events	1449
16.6.2. List stack events	1450
16.6.3. List resource events	1453
16.6.4. Show event details	1456
16.7. Templates	1457
16.7.1. Get stack template	1458
16.7.2. List template versions	1460
16.7.3. Validate template	1461
16.7.4. Show resource template	1463
16.7.5. Show resource schema	1465
16.7.6. List resource types	1467
16.8. Build info	1468

16.8.1. Show build information	1469
16.9. Software configuration	1469
16.9.1. Create configuration	1471
16.9.2. Show configuration details	1473
16.9.3. Delete config	1475
16.9.4. List deployments	1476
16.9.5. Create deployment	1478
16.9.6. Show server configuration metadata	1479
16.9.7. Show deployment details	1483
16.9.8. Update deployment	1484
16.9.9. Delete deployment	1486
16.10. Manage service	1486
16.10.1. Show orchestration engine status	1487
17. Shared File Systems API v2 (CURRENT)	1489
17.1. API versions	1492
17.1.1. List versions	1493
17.2. API extensions	1494
17.2.1. List extensions	1495
17.3. Limits	1497
17.3.1. Get share limits	1498
17.4. Shares	1498
17.4.1. Create share	1501
17.4.2. List shares	1503
17.4.3. List shares (detailed)	1506
17.4.4. Show share	1510
17.4.5. Update share	1512
17.4.6. Delete share	1514
17.5. Share metadata	1514
17.5.1. Show share metadata	1515
17.5.2. Update share metadata	1516
17.5.3. Set share metadata	1517
17.5.4. Unset share metadata	1518
17.6. Share actions	1518
17.6.1. Grant access	1520
17.6.2. Revoke access	1521
17.6.3. List access rules	1522
17.6.4. Reset share state	1523
17.6.5. Force-delete share	1524
17.6.6. Extend share	1525
17.6.7. Shrink share	1526
17.7. Share snapshots	1526
17.7.1. Create share snapshot	1528
17.7.2. List share snapshots	1530
17.7.3. List share snapshots (detailed)	1532
17.7.4. Show share snapshot details	1534
17.7.5. Update share snapshot	1536
17.7.6. Delete share snapshot	1538
17.7.7. Reset share snapshot state	1539
17.7.8. Force-delete share snapshot	1540
17.8. Share networks	1540
17.8.1. Create share network	1542

17.8.2. List share networks	1543
17.8.3. List share networks with details	1544
17.8.4. Show share network	1546
17.8.5. Update share network	1547
17.8.6. Delete share network	1549
17.8.7. Add security service to share network	1550
17.8.8. Remove security service from share network	1551
17.9. Security services	1552
17.9.1. Create security service	1553
17.9.2. List security services	1555
17.9.3. List security services with details	1556
17.9.4. Show security service	1558
17.9.5. Update security service	1559
17.9.6. Delete security service	1561
17.10. Share servers	1561
17.10.1. List share servers	1563
17.10.2. Show share server	1564
17.10.3. Delete share server	1565
17.10.4. Show share server details	1566
17.11. Share instances (since API v2.3)	1566
17.11.1. List share instances	1568
17.11.2. Show share instance	1570
17.11.3. Reset share instance state	1571
17.11.4. Force-delete share instance	1572
17.12. Share types	1572
17.12.1. Create share type	1575
17.12.2. List share types	1577
17.12.3. List default share types	1579
17.12.4. Delete share type	1580
17.12.5. Add share type access	1581
17.12.6. Remove share type access	1582
17.12.7. Show share type access	1583
17.12.8. List extra specs	1584
17.12.9. Set extra spec	1585
17.12.10. Unset an extra spec	1586
17.13. Back-end storage pools	1586
17.13.1. List back-end storage pools	1587
17.13.2. List back-end storage pools with details	1588
17.14. Services	1589
17.14.1. List services	1590
17.14.2. Enable service	1591
17.14.3. Disable service	1592
17.15. Availability zones	1592
17.15.1. List availability zones	1593
17.16. Manage share	1593
17.16.1. Manage share	1594
17.17. Unmanage share	1595
17.17.1. Unmanage share	1596
17.18. Quota sets	1596
17.18.1. Show quotas	1597
17.18.2. Update quotas	1598

17.18.3. Delete quotas	1599
17.18.4. Show default quotas	1600
18. Shared File Systems API v1 (SUPPORTED)	1601
19. Telemetry API v2 (CURRENT)	1602
19.1. Alarms	1602
19.1.1. List alarms	1604
19.1.2. Create alarm	1606
19.1.3. Show alarm details	1608
19.1.4. Update alarm	1610
19.1.5. Delete alarm	1612
19.1.6. Update alarm state	1613
19.1.7. Show alarm state	1614
19.1.8. Show alarm history	1615
19.2. Meters	1615
19.2.1. List meters	1618
19.2.2. Add samples to meter	1620
19.2.3. List samples for meter	1623
19.2.4. Show meter statistics	1625
19.3. Samples	1626
19.3.1. List samples	1627
19.3.2. Show sample details	1629
19.4. Resources	1630
19.4.1. List resources	1631
19.4.2. Show resource details	1633
19.5. Capabilities	1634
19.5.1. List capabilities	1635

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.
Volume type access (volumes)		
POST	/v2/{tenant_id}/volumes	Creates a private Block Storage volume.
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.
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 be detached. 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 os-reenable-replica action in the request body.

Method	URI	Description
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 <code>os-force_delete</code> 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/{tenant_id}/os-quota-sets/{tenant_id}{?usage}	Shows quotas for a tenant.
PUT	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.
DELETE	/v2/{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/defaults	Gets default quotas for a tenant.
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a tenant and user.
PUT	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a tenant and user.
DELETE	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Deletes quotas for a user so that the quotas revert to default values.
GET	/v2/{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	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 or unsets 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{?vol_type_id}	Associates a QoS specification with a volume type.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate{?vol_type_id}	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	Gets all associations for a QoS specification.
Volume types (types)		

Method	URI	Description
GET	/v2/{tenant_id}/types	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{?snapshot,volume_id,force,name,description}	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 that is associated with 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.
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	Lists consistency groups.
POST	/v2/{tenant_id}/consistencygroups	Creates a consistency group.

Method	URI	Description
GET	/v2/{tenant_id}/consistency-groups/detail	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}/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: 200, 300

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": [
        {
            "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"
        }
    ]
}
```

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 1.2. Show API v2 details: JSON response

```
{
  "version": {
    "id": "v2.0",
    "links": [
      {
        "href": "http://23.253.211.234:8776/v2/v2.0",
        "rel": "self"
      }
    ],
    "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": "CURRENT"
  }
}
```

Example 1.3. Show API v2 details: JSON response

```
{
  "version": {
    "id": "v2.0",
    "links": [
      {
        "href": "http://23.253.211.234:8776/v2/v2.0",
        "rel": "self"
      }
    ],
    "media-types": [
      {
        "base": "application/xml",
        "type": "application/vnd.openstack.volume+xml;version=1"
      },
      {
        "base": "application/json",
        "type": "application/vnd.openstack.volume+json;version=1"
      }
    ]
  }
}
```

```
        "type": "application/vnd.openstack.volume+json;version=1"
    }
],
"status": "CURRENT"
}
```

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: 200, 300

1.2.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

1.2.1.2. Response

Example 1.4. 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": "Expose the internal project_id as an attribute of a volume."
        }
    ]
}
```

```
        "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": []
    }
]
```

```
        "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."
}
]
}

```

Example 1.5. 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>
    <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}	String	The ID for the tenant or project in a multi-tenancy cloud.

This operation does not accept a request body.

1.3.1.2. Response

Example 1.6. Show absolute limits: JSON response

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

Example 1.7. 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="maxTotalVolumeGigabytes" value="1000"/>
        <limit name="totalGigabytesUsed" value="0"/>
        <limit name="maxTotalSnapshots" value="10"/>
        <limit name="totalVolumesUsed" value="0"/>
        <limit name="maxTotalVolumes" value="10"/>
    </absolute>
</limits>
```

Example 1.8. Show absolute limits: JSON response

```
{
```

```

    "limits": {
        "rate": [],
        "absolute": {
            "totalSnapshotsUsed": 0,
            "maxTotalVolumeGigabytes": 1000,
            "totalGigabytesUsed": 0,
            "maxTotalSnapshots": 10,
            "totalVolumesUsed": 0,
            "maxTotalVolumes": 10
        }
    }
}

```

Example 1.9. 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="maxTotalVolumeGigabytes" value="1000"/>
        <limit name="totalGigabytesUsed" value="0"/>
        <limit name="maxTotalSnapshots" value="10"/>
        <limit name="totalVolumesUsed" value="0"/>
        <limit name="maxTotalVolumes" value="10"/>
    </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_volid` 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 be attached 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 during volume creation.
error_deleting	An error occurred during volume deletion.
backing-up	The volume is being backed up.
restoring-backup	A backup is being restored to the volume.

Status	Description
error_restoring	An error occurred during backup restoration to a volume.
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.

1.4.1. Create volume

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

To create a bootable volume, include the ID 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 which 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}	String	The unique identifier of the tenant or account.

Example 1.10. Create volume: JSON request

```
{
  "volume": {
    "status": "creating",
    "description": null,
    "availability_zone": null,
    "source_volid": null,
    "consistencygroup_id": null,
    "snapshot_id": null,
    "source_replica": null,
    "size": 10,
    "user_id": null,
  }
}
```

```

        "name": null,
        "imageRef": null,
        "attach_status": "detached",
        "volume_type": null,
        "project_id": null,
        "metadata": {}
    }
}

```

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

Name	Type	Description
availability_zone	String <i>(Optional)</i>	The availability zone.
source_volid	Uuid <i>(Optional)</i>	To create a volume from an existing volume, specify the ID of the existing volume. The volume is created with the same size as the source volume.
description	String <i>(Optional)</i>	The volume description.
snapshot_id	Uuid <i>(Optional)</i>	To create a volume from an existing snapshot, specify the ID of the existing volume snapshot. The volume is created in same availability zone and with same size as the snapshot.
size	Int <i>(Required)</i>	The size of the volume, in gibibytes (GiB).
name	String <i>(Optional)</i>	The volume name.
imageRef	Uuid <i>(Optional)</i>	The ID of the image from which you want to create the volume. Required to create a bootable volume.
volume_type	String <i>(Optional)</i>	The associated volume type.
metadata	String <i>(Optional)</i>	One or more metadata key and value pairs to associate with the volume.

Example 1.11. 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 1.12. Create volume: JSON response

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

```

    "links": [
        {
            "href": "http://volume.example.com:8776/v2/{tenant_id}/
volumes/{volume_id}",
            "rel": "self"
        },
        {
            "href": "http://volume.example.com:8776/{tenant_id}/volumes/
{volume_id}",
            "rel": "bookmark"
        }
    ],
    "availability_zone": "nova",
    "bootable": "false",
    "encrypted": false,
    "created_at": "2015-05-17T18:14:34.000000",
    "description": null,
    "os-vol-tenant-attr:tenant_id": "{tenant_id}",
    "os-volume-replication:driver_data": null,
    "volume_type": "lvmdriver-1",
    "name": null,
    "replication_status": "disabled",
    "consistencygroup_id": null,
    "source_volid": null,
    "snapshot_id": null,
    "os-volume-replication:extended_status": null,
    "metadata": {},
    "id": "7449e332-9715-4185-b191-1cf91d75d48",
    "size": 10
}
}

```

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

Name	Type	Description
status	String <i>(Required)</i>	The volume status.
name	String <i>(Required)</i>	The volume name.
attachments	Dict <i>(Required)</i>	One or more instance attachments.
availability_zone	String <i>(Required)</i>	The availability zone.
created_at	Datetime <i>(Required)</i>	Date and time when the volume was created.
description	String <i>(Required)</i>	The volume description.
volume_type	String <i>(Required)</i>	The associated volume type.
snapshot_id	Uuid <i>(Required)</i>	The ID of the source volume snapshot.
source_volid	Uuid	The ID of the source volume.

Name	Type	Description
	(Required)	
metadata	String (Required)	One or more metadata key and value pairs to associate with the volume.
id	Uuid (Required)	The volume ID.
size	Int (Required)	The size of the volume, in gibibytes (GiB).
os-vol-volume-replication:driver_data	String (Optional)	The name of the volume replication driver.
os-vol-volume-replication:extended_status	String (Optional)	The status of the volume replication.

Example 1.13. 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}	String	The unique identifier of the tenant or account.

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 <code>asc</code> (ascending) or <code>desc</code> (descending).
limit	Int (Optional)	Requests a page size of returned items from the query. 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)	Specifies 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.

1.4.2.2. Response

Example 1.14. 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 1.15. 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}	String	The unique identifier of the tenant or account.

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

Name	Type	Description
sort	String <i>(Optional)</i>	Comma-separated list of sort keys and optional sort directions in the form of <key>[:<direction>]. A valid direction is <code>asc</code> (ascending) or <code>desc</code> (descending).
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

1.4.3.2. Response

Example 1.16. 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_volid="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 1.17. List volumes with details: JSON response

```
{
  "volumes": [
    {
      "status": "available",
      "attachments": [],
      "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"
        }
      ],
      "availability_zone": "nova",
      "os-vol-host-attr:host": "ip-10-168-107-25",
      "source_volid": null,
      "snapshot_id": null,
      "id": "45baf976-c20a-4894-a7c3-c94b7376bf55",
      "description": "Another volume.",
      "name": "vol-004",
      "created_at": "2013-02-25T06:36:28.000000",
      "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": "junk"
      }
    },
    {
      "status": "available",
      "attachments": [],
      "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"
        }
    ],
    "availability_zone": "nova",
    "os-vol-host-attr:host": "ip-10-168-107-25",
    "source_volid": null,
    "snapshot_id": null,
    "id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
    "description": "This is yet, another volume.",
    "name": "vol-003",
    "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.

1.4.4. 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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

This operation does not accept a request body.

1.4.4.2. Response

Example 1.18. 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 1.19. 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/
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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

Example 1.20. 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 1.21. 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 1.22. 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 1.23. Update volume: 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/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
            "rel": "bookmark"
        }
    ],
    "availability_zone": "nova",
    "source_volid": null,
    "snapshot_id": null,
    "id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
    "description": "This is yet, another volume.",
    "name": "vol-003",
    "created_at": "2013-02-25T02:40:21.000000",
    "volume_type": "None",
    "size": 1,
    "metadata": {
        "contents": "not junk"
    }
}
```

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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

This operation does not accept a request body.

1.5. Volume type access (volumes)

Creates private volumes.

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

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

1.5.1. Create private volume

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

Normal response codes: 202

1.5.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.

Example 1.24. Create private volume: JSON request

```
{
    "volume": {
        "availability_zone": null,
        "source_volid": null,
        "description": null,
        "snapshot_id": null,
        "size": 10,
        "name": "my_volume",
        "imageRef": null,
        "volume_type": null,
        "metadata": {},
        "os-volume-type-access: is_public": false
    }
}
```

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

Name	Type	Description
availability_zone	String <i>(Optional)</i>	The availability zone.
source_volid	Uuid <i>(Optional)</i>	To create a volume from an existing volume, specify the ID of the existing volume. The volume is created with the same size as the source volume.
description	String <i>(Optional)</i>	The volume description.
snapshot_id	Uuid <i>(Optional)</i>	To create a volume from an existing snapshot, specify the ID of the existing volume snapshot. The volume is created in same availability zone and with same size as the snapshot.
size	Int <i>(Required)</i>	The size of the volume, in gibibytes (GiB).
name	String <i>(Optional)</i>	The volume name.
imageRef	Uuid <i>(Optional)</i>	The ID of the image from which you want to create the volume. Required to create a bootable volume.
volume_type	String	The associated volume type.

Name	Type	Description
	(Optional)	
metadata	String (Optional)	One or more metadata key and value pairs to associate with the volume.
os-volume-type-access:is_public	Boolean (Optional)	The volume type access. Set to true if access is public. Set to false if access is private. Default is true.

Example 1.25. Create private volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
  xmlns:os-volume-type-access="http://docs.openstack.org/openstack-block-
storage/2.0/ext/os-volume-type-access/api/v2.0" name="vol-001" description=
"Another volume." size="2" os-volume-type-access:is_public="false" />
```

This operation does not accept a request body.

1.5.1.2. Response

Example 1.26. Create private volume: JSON response

```
{
  "volume": {
    "status": "creating",
    "name": "my_volume",
    "attachments": [],
    "availability_zone": "nova",
    "bootable": "false",
    "created_at": "2014-02-21T19: 52: 04.949734",
    "description": null,
    "volume_type": "None",
    "snapshot_id": null,
    "source_volid": null,
    "metadata": {},
    "id": "93c2e2aa-7744-4fd6-a31a-80c4726b08d7",
    "size": 10,
    "os-volume-type-access: is_public": false
  }
}
```

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

Name	Type	Description
status	String (Required)	The volume status.
name	String (Required)	The volume name.
attachments	Dict (Required)	One or more instance attachments.
availability_zone	String (Required)	The availability zone.

Name	Type	Description
created_at <i>(Required)</i>	Datetime	Date and time when the volume was created.
description <i>(Required)</i>	String	The volume description.
volume_type <i>(Required)</i>	String	The associated volume type.
snapshot_id <i>(Required)</i>	Uuid	The ID of the source volume snapshot.
source_volid <i>(Required)</i>	Uuid	The ID of the source volume.
metadata <i>(Required)</i>	String	One or more metadata key and value pairs to associate with the volume.
id <i>(Required)</i>	Uuid	The volume ID.
size <i>(Required)</i>	Int	The size of the volume, in gibibytes (GiB).
os-vol-volume-replication:driver_data <i>(Optional)</i>	String	The name of the volume replication driver.
os-vol-volume-replication:extended_status <i>(Optional)</i>	String	The status of the volume replication.
os-volume-type-access:is_public <i>(Required)</i>	Boolean	The volume type access. Set to <code>true</code> if access is public. Set to <code>false</code> if access is private.

Example 1.27. Create private volume: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
  xmlns:atom="http://www.w3.org/2005/Atom" xmlns:os-volume-type-access=
  "http://docs.openstack.org/openstack-block-storage/2.0/ext/os-volume-type-
  access/api/v2.0" 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" os-volume-type-
  access:is_public="false">
  <attachments />
  <metadata />
</volume>
```

This operation does not return a response body.

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 backend 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 <code>os-extend</code> 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 <code>os-reset_status</code> action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Sets the image metadata for a volume. Specify the <code>os-set_image_metadata</code> action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Removes image metadata, by key, from a volume. Specify the <code>os-unset_image_metadata</code> 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 <code>os-attach</code> 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 <code>os-unmanage</code> action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Forces a volume to be detached. Specify the <code>os-force_detach</code> action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Promotes a replicated volume. Specify the <code>os-promote-replica</code> action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Re-enables replication of a volume. Specify the <code>os-reenable-replica</code> 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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

Example 1.28. 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 <code>os-reset_status</code> 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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

Example 1.29. 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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

Example 1.30. 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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

Example 1.31. 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 <code>os-attach</code> 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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

Example 1.32. Attach volume to server: JSON request

```
{
    "os-attach": {
        "instance_uuid": "95D9EF50-507D-11E5-B970-0800200C9A66",
        "host_name": "cinder-2",
        "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 <code>os-unmanage</code> 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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

Example 1.33. 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 be detached. 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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

Example 1.34. 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 <code>os-promote-replica</code> 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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

Example 1.35. 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 os-reenable-replica 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}	String	The unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

Example 1.36. 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 be restored to a volume.
deleting	The backup is being deleted.
error	An error has occurred with the backup.
restoring	The backup is being restored to a volume.
error_restoring	An error occurred during backup restoration to a volume.

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}	String	The unique identifier of the tenant or account.

Example 1.37. 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 1.38. 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/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}	String	The unique identifier of the tenant or account.

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

Name	Type	Description
sort_key	String <i>(Optional)</i>	Sorts by the requested image attribute. Accepted values are name, status, container_format, disk_format, size, id, created_at, and updated_at. Default is created_at. The API uses the natural sorting direction of the image attribute that is provided as the sort_key.
sort_dir	String <i>(Optional)</i>	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 <i>(Optional)</i>	Requests a page size of returned items from the query. 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 <i>(Optional)</i>	Specifies 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 1.39. 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}	String	The unique identifier of the tenant or account.

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

Name	Type	Description
sort_key	String <i>(Optional)</i>	Sorts by the requested image attribute. Accepted values are name, status, container_format, disk_format, size, id, created_at, and updated_at. Default is created_at. The API uses the natural sorting direction of the image attribute that is provided as the sort_key.
sort_dir	String <i>(Optional)</i>	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 <i>(Optional)</i>	Requests a page size of returned items from the query. 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 <i>(Optional)</i>	Specifies 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 1.40. 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}	String	The unique identifier of the tenant or account.
{backup_id}	UUID	The unique identifier for a backup.

This operation does not accept a request body.

1.7.4.2. Response

Example 1.41. 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/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: 202

1.7.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{backup_id}	UUID	The unique identifier for a backup.

This operation does not accept a request body.

1.7.6. Restore backup

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

You must specify either the volume ID or name. If you specify both the volume ID and name, the ID 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}	String	The unique identifier of the tenant or account.
{backup_id}	UUID	The unique identifier for a backup.

Example 1.42. Restore backup: JSON request

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

1.7.6.2. Response

Example 1.43. 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}/back-ups/{backup_id}/action	Force-deletes a backup. Specify the <code>os-force_delete</code> action in the request body.

1.8.1. Force-delete backup

Method	URI	Description
POST	/v2/{tenant_id}/back-ups/{backup_id}/action	Force-deletes a backup. Specify the <code>os-force_delete</code> 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}	String	The unique identifier of the tenant or account.
{backup_id}	UUID	The unique identifier for a backup.

Example 1.44. 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}	String	The unique identifier of the tenant or account.
{hostname}	String	The name of the host that hosts the storage back end.

1.9.1.2. Response

Example 1.45. 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/{tenant_id}/os-quota-sets/{tenant_id}{?usage}	Shows quotas for a tenant.
PUT	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.
DELETE	/v2/{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/defaults	Gets default quotas for a tenant.
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a tenant and user.
PUT	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a tenant and user.
DELETE	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Deletes quotas for a user so that the quotas revert to default values.
GET	/v2/{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/{tenant_id}/os-quota-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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID for the tenant for which you want to show, update, or delete quotas. This ID is different than the first tenant ID that you specify in the URI: That ID is for the admin tenant.

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

Name	Type	Description
usage	Boolean <i>(Optional)</i>	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 1.46. 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	String <i>(Required)</i>	A quota_set object.
gigabytes	Int <i>(Required)</i>	The number of gigabytes allowed for each tenant.
snapshots	Int <i>(Required)</i>	The number of snapshots allowed for each tenant.
volumes	Int <i>(Required)</i>	The number of volumes allowed for each tenant.

Name	Type	Description
in_use	String <i>(Optional)</i>	The in use data size. Visible only if you set the <code>usage=True</code> query parameter.
reserved	Int <i>(Optional)</i>	Reserved volume size. Visible only if you set the <code>usage=True</code> query parameter.

Example 1.47. 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/{tenant_id}/os-quota-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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID for the tenant for which you want to show, update, or delete quotas. This ID is different than the first tenant ID that you specify in the URI: That ID is for the admin tenant.

Example 1.48. Update quotas response: JSON

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

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

Name	Type	Description
quota_set	String <i>(Required)</i>	A quota_set object.
gigabytes	Int <i>(Required)</i>	The number of gigabytes allowed for each tenant.
snapshots	Int <i>(Required)</i>	The number of snapshots allowed for each tenant.
volumes	Int <i>(Required)</i>	The number of volumes allowed for each tenant.

Example 1.49. 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 1.50. Update quota response: JSON

```
{
```

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

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

Name	Type	Description
quota_set	String <i>(Required)</i>	A quota_set object.
gigabytes	Int <i>(Required)</i>	The number of gigabytes allowed for each tenant.
snapshots	Int <i>(Required)</i>	The number of snapshots allowed for each tenant.
volumes	Int <i>(Required)</i>	The number of volumes allowed for each tenant.

Example 1.51. 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/{tenant_id}/os-quota-sets/{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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID for the tenant for which you want to show, update, or delete quotas. This ID is different than the first tenant ID that you specify in the URI: That ID is for the admin 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/defaults	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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

1.10.4.2. Response

Example 1.52. 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	String (Required)	A quota_set object.
gigabytes	Int (Required)	The number of gigabytes allowed for each tenant.
snapshots	Int (Required)	The number of snapshots allowed for each tenant.
volumes	Int (Required)	The number of volumes allowed for each tenant.

Example 1.53. 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/{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

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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID for the tenant for which you want to show or update quotas. This ID is different than the first tenant ID that you specify in the URI: That ID is for the admin tenant.
{user_id}	String	The user ID. Specify in the URI as a query string: user_id={user_id}.

This operation does not accept a request body.

1.10.5.2. Response

Example 1.54. 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	String <i>(Required)</i>	A quota_set object.
gigabytes	Int <i>(Required)</i>	The number of gigabytes allowed for each tenant.
snapshots	Int <i>(Required)</i>	The number of snapshots allowed for each tenant.
volumes	Int <i>(Required)</i>	The number of volumes allowed for each tenant.

Example 1.55. 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/{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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID for the tenant for which you want to show or update quotas. This ID is different than the first tenant ID that you specify in the URI: That ID is for the admin tenant.
{user_id}	String	The user ID. Specify in the URI as a query string: user_id={user_id}.

Example 1.56. 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	String <i>(Required)</i>	A quota_set object.
gigabytes	Int <i>(Required)</i>	The number of gigabytes allowed for each tenant.
snapshots	Int <i>(Required)</i>	The number of snapshots allowed for each tenant.
volumes	Int <i>(Required)</i>	The number of volumes allowed for each tenant.

Example 1.57. 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 1.58. 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	String <i>(Required)</i>	A quota_set object.
gigabytes	Int <i>(Required)</i>	The number of gigabytes allowed for each tenant.
snapshots	Int <i>(Required)</i>	The number of snapshots allowed for each tenant.
volumes	Int <i>(Required)</i>	The number of volumes allowed for each tenant.

Example 1.59. 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/{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

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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID for the tenant for which you want to show or update quotas. This ID is different than the first tenant ID that you specify in the URI: That ID is for the admin tenant.
{user_id}	String	The user ID. Specify in the URI as a query string: 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/{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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID for the tenant for which you want to show or update quotas. This ID is different than the first tenant ID that you specify in the URI: That ID is for the admin tenant.
{user_id}	String	The user ID. Specify in the URI as a query string: user_id={user_id}.

This operation does not accept a request body.

1.10.8.2. Response

Example 1.60. 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	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 or unsets 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{?vol_type_id}	Associates a QoS specification with a volume type.

Method	URI	Description
GET	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate{?vol_type_id}	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	Gets 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.

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}	String	The unique ID of the tenant or account.

Example 1.61. 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	String <i>(Required)</i>	A qos_specs object.
name	String <i>(Required)</i>	The name of the QoS specification.
specs	String <i>(Required)</i>	Specification key and value pairs.

Example 1.62. 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 1.63. Create 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 table shows the body parameters for the create qos specification response:

Name	Type	Description
qos_specs	String <i>(Required)</i>	A qos_specs object.
specs	String <i>(Required)</i>	A specs object.
consumer	String <i>(Required)</i>	The consumer type.
name	String <i>(Required)</i>	The name of the QoS specification.
id	Uuid <i>(Required)</i>	The generated ID for the QoS specification.
links	Dict <i>(Required)</i>	The QoS specification links.

Example 1.64. Create QoS specification: XML response

```

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

1.11.2. List QoS specs

Method	URI	Description
GET	/v2/{tenant_id}/qos-specs	Lists quality of service (QoS) specifications.

Normal response codes: 200, 300

1.11.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique ID of the tenant or account.

This operation does not accept a request body.

1.11.2.2. Response

Example 1.65. 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	String <i>(Required)</i>	A qos_specs object.
specs	String <i>(Required)</i>	Specification key and value pairs.
consumer	String <i>(Required)</i>	The consumer type.

Name	Type	Description
name	String <i>(Required)</i>	The name of the QoS specification.
id	Uuid	The generated ID for the QoS specification. <i>(Required)</i>

Example 1.66. 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}	String	The unique ID of the tenant or account.
{qos_id}	UUID	The UUID for the QoS specification.

This operation does not accept a request body.

1.11.3.2. Response

Example 1.67. 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/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	String	A qos_specs object.

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

Example 1.68. 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 or unset keys in QoS specification

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

Normal response codes: 200

1.11.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique ID of the tenant or account.
{qos_id}	UUID	The UUID for the QoS specification.

Example 1.69. Set or unset keys in QoS specification: XML request

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

Example 1.70. Set or unset keys in QoS specification: JSON request

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

This operation does not accept a request body.

1.11.4.2. Response

Example 1.71. Set or unset keys in QoS specification: XML response

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

Example 1.72. Set or unset 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}	String	The unique ID of the tenant or account.
{qos_id}	UUID	The UUID for the QoS specification.
{qos_id}	String	The unique ID of the QoS specification.
{force}	String	Optional flag that indicates whether to delete a QoS specification, even if it is in-use.

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{?vol_type_id}	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}	String	The unique ID of the tenant or account.
{qos_id}	UUID	The UUID for the QoS specification.

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{?vol_type_id}	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}	String	The unique ID of the tenant or account.
{qos_id}	UUID	The UUID for the QoS specification.

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}	String	The unique ID of the tenant or account.
{qos_id}	UUID	The UUID for 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	Gets 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}	String	The unique ID of the tenant or account.
{qos_id}	UUID	The UUID for the QoS specification.

This operation does not accept a request body.

1.11.9.2. Response

Example 1.73. Get all associations for QoS specification: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<qos_associations>
  <associations association_type="volume_type"
    name="reliability-type"
    id="a12983c2-83bd-4afa-be9f-ad796573ead6" />
</qos_associations>
```

Example 1.74. Get all associations for QoS specification: JSON response

```
{
  "qos_associations": [
    {
      "association_type": "volume_type",
      "name": "reliability-type",
      "id": "a12983c2-83bd-4afa-be9f-ad796573ead6"
    }
  ]
}
```

This operation does not return a response body.

1.12. Volume types (types)

Method	URI	Description
GET	/v2/{tenant_id}/types	Lists volume types.
POST	/v2/{tenant_id}/types	Creates a volume type.
PUT	/v2/{tenant_id}/types/{volume_type_id}	Updates 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.
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	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}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

1.12.1.2. Response

Example 1.75. 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_types>
```

Example 1.76. 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.

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}	String	The unique identifier of the tenant or account.

Example 1.77. 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 1.78. Create volume type: JSON request

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

This operation does not accept a request body.

1.12.2.2. Response

Example 1.79. 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">
    <extra_specs>
        <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
</volume_type>
```

Example 1.80. Create volume type: JSON response

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

```
        "id": "6685584b-1eac-4da6-b5c3-555430cf68ff",
        "name": "vol-type-001",
        "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.

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}	String	The unique identifier of the tenant or account.
{volume_type_id}	UUID	The unique identifier for an existing volume type.

Example 1.81. 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 1.82. Update volume type: JSON request

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

This operation does not accept a request body.

1.12.3.2. Response

Example 1.83. 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 1.84. 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.

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}	String	The unique identifier of the tenant or account.
{volume_type_id}	UUID	The unique identifier for an existing volume type.

Example 1.85. 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 1.86. 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.

1.12.4.2. Response

Example 1.87. 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 1.88. 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.

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}	String	The unique identifier of the tenant or account.
{volume_type_id}	UUID	The unique identifier for an existing volume type.

This operation does not accept a request body.

1.12.5.2. Response

Example 1.89. Show volume type details: JSON response

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

Example 1.90. 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">
    <extra_specs>
        <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
</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}	String	The unique identifier of the tenant or account.
{volume_type_id}	UUID	The unique identifier 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 be used.
deleting	The snapshot is being deleted.
error	An error occurred during snapshot creation.
error_deleting	An error occurred during snapshot deletion.

Method	URI	Description
POST	/v2/{tenant_id}/snapshots{?snapshot,volume_id,force,name,description}	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{?snapshot, volume_id, force, name, description}	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}	String	The unique identifier of the tenant or account.

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

Name	Type	Description
snapshot	String <i>(Required)</i>	A partial representation of a snapshot used in the creation process.
volume_id	String <i>(Required)</i>	To create a snapshot from an existing volume, specify the ID of the existing volume.
force	Boolean <i>(Optional)</i>	[True/False] Indicate whether to snapshot, even if the volume is attached. Default==False.
name	String <i>(Optional)</i>	Name of the snapshot. Default==None.
description	String <i>(Optional)</i>	Description of snapshot. Default==None.

Example 1.91. 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 1.92. 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 1.93. 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 1.94. 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}	String	The unique identifier of the tenant or account.

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

Name	Type	Description
sort_key	String <i>(Optional)</i>	Sorts by the requested image attribute. Accepted values are name, status, container_format, disk_format, size, id, created_at, and updated_at. Default is created_at. The API uses the natural sorting direction of the image attribute that is provided as the sort_key.
sort_dir	String <i>(Optional)</i>	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 <i>(Optional)</i>	Requests a page size of returned items from the query. 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 <i>(Optional)</i>	Specifies 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.

1.13.2.2. Response

Example 1.95. List snapshots: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshots>
    <snapshot 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">
        <metadata/>
    </snapshot>
    <snapshot status="available" description="Weekly backup"
        created_at="2013-02-25 07:20:38"
        volume_id="806092e3-7551-4fff-a005-49016f4943b1" size="1"
        id="e820db06-58b5-439d-bac6-c01faa3f6499" name="snap-002">
        <metadata/>
```

```
</snapshot>
</snapshots>
```

Example 1.96. List snapshots: JSON response

```
{
  "snapshots": [
    {
      "status": "available",
      "description": "Very important",
      "created_at": "2013-02-25T04:13:17.000000",
      "metadata": {},
      "volume_id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
      "size": 1,
      "id": "2bb856e1-b3d8-4432-a858-09e4ce939389",
      "name": "snap-001"
    },
    {
      "status": "available",
      "description": "Weekly backup",
      "created_at": "2013-02-25T07:20:38.000000",
      "metadata": {},
      "volume_id": "806092e3-7551-4fff-a005-49016f4943b1",
      "size": 1,
      "id": "e820db06-58b5-439d-bac6-c01faa3f6499",
      "name": "snap-002"
    }
  ]
}
```

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}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

1.13.3.2. Response

Example 1.97. List snapshots with details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshots
    xmlns:os-extended-snapshot-attributes="http://docs.openstack.org/
openstack-block-storage/2.0/content/Extended_Snapshot_Attributes.html">
    <snapshot status="available" description="Daily backup"
        created_at="2013-02-25 07:30:12"
        volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="30"
        id="43f20e0e-2c2c-4770-9d4e-c3d769ae5470" name="snap-001"
        os-extended-snapshot-attributes:project_id=
"0c2eba2c5af04d3f9e9d0d410b371fde"
        os-extended-snapshot-attributes:progress="100%">
        <metadata/>
    </snapshot>
    <snapshot status="available" description="Weekly backup"
        created_at="2013-02-25 07:20:38"
        volume_id="806092e3-7551-4fff-a005-49016f4943b1" size="1"
        id="e820db06-58b5-439d-bac6-c01faa3f6499" name="snap-002"
        os-extended-snapshot-attributes:project_id=
"0c2eba2c5af04d3f9e9d0d410b371fde"
        os-extended-snapshot-attributes:progress="100%">
        <metadata/>
    </snapshot>
</snapshots>
```

Example 1.98. List snapshots with details: JSON response

```
{
    "snapshots": [
        {
            "status": "available",
            "os-extended-snapshot-attributes:progress": "100%",
            "description": "Daily backup",
            "created_at": "2013-02-25T07:30:12.000000",
            "metadata": {} ,
```

```
        "volume_id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
        "os-extended-snapshot-attributes:project_id":
"0c2eba2c5af04d3f9e9d0d410b371fde",
        "size": 30,
        "id": "43f20e0e-2c2c-4770-9d4e-c3d769ae5470",
        "name": "snap-001"
    },
    {
        "status": "available",
        "os-extended-snapshot-attributes:progress": "100%",
        "description": "Weekly backup",
        "created_at": "2013-02-25T07:20:38.000000",
        "metadata": {},
        "volume_id": "806092e3-7551-4fff-a005-49016f4943b1",
        "os-extended-snapshot-attributes:project_id":
"0c2eba2c5af04d3f9e9d0d410b371fde",
        "size": 1,
        "id": "e820db06-58b5-439d-bac6-c01faa3f6499",
        "name": "snap-002"
    }
]
```

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}	String	The unique identifier of the tenant or account.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

1.13.4.2. Response

Example 1.99. 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 1.100. 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}	String	The unique identifier of the tenant or account.
{snapshot_id}	UUID	The UUID of the snapshot.

Example 1.101. 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 1.102. 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 1.103. Update snapshot: 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="This is yet, another snapshot"
    created_at="2013-02-20T08:11:34.000000"
    volume_id="2402b902-0b7a-458c-9c07-7435a826f794"
    size="1"
    id="4b502fcf-1f26-45f8-9fe5-3b9a0a52eaf2"
    name="snap-002"
    os-extended-snapshot-attributes:project_id=
    "0c2eba2c5af04d3f9e9d0d410b371fde"
    os-extended-snapshot-attributes:progress="100%">
    <metadata/>
</snapshot>
```

Example 1.104. 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}	String	The unique identifier of the tenant or account.
{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}	String	The unique identifier of the tenant or account.
{snapshot_id}	UUID	The UUID of the snapshot.

This operation does not accept a request body.

1.13.7.2. Response

Example 1.105. 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">v2</meta>
    </metadata>
</snapshot>
```

Example 1.106. 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": "v2"
        },
        "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.

1.13.8. Update snapshot metadata

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

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}	String	The unique identifier of the tenant or account.
{snapshot_id}	UUID	The UUID of the snapshot.

Example 1.107. Update snapshot metadata: XML request

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

Example 1.108. Update snapshot metadata: JSON request

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

This operation does not accept a request body.

1.13.8.2. Response

Example 1.109. 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 1.110. 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}	String	The unique identifier of the tenant or account.

Example 1.111. 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 1.112. 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/
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 that is associated with a volume.

1.15.1. Show image metadata for volume

Method	URI	Description
GET	/v2/{tenant_id}/os-vol-image-meta	Shows image metadata that is associated with a volume.

When the request is made, the caller must specify (in the `ref` element) a reference to an existing storage volume. 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}	String	The unique identifier of the tenant or account.

Example 1.113. 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 1.114. 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}	String	The unique identifier of the tenant or account.

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

Name	Type	Description
detail	Boolean <i>(Optional)</i>	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 1.115. 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"
      }
    }
  ]
}
```

```
        "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}	String	The unique identifier of the tenant or account.

Example 1.116. Create volume transfer: JSON request

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

1.17.1.2. Response

Example 1.117. 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}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

1.17.2.2. Response

Example 1.118. 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}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

1.17.3.2. Response

Example 1.119. 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}	String	The unique identifier of the tenant or account.
{transfer_id}	UUID	The unique identifier for a volume transfer.

This operation does not accept a request body.

1.17.4.2. Response

Example 1.120. 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}	String	The unique identifier of the tenant or account.
{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}	String	The unique identifier of the tenant or account.
{transfer_id}	UUID	The unique identifier for a volume transfer.

Example 1.121. Accept volume transfer: JSON request

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

1.17.6.2. Response

Example 1.122. 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	Lists consistency groups.
POST	/v2/{tenant_id}/consistencygroups	Creates a consistency group.
GET	/v2/{tenant_id}/consistency-groups/detail	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}/update	Updates a consistency group.

1.18.1. List consistency groups

Method	URI	Description
GET	/v2/{tenant_id}/consistencygroups	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}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

1.18.1.2. Response

Example 1.123. List consistency groups: JSON response

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

This operation does not return a response body.

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}	String	The unique identifier of the tenant or account.

Example 1.124. 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"
  }
}
```

This operation does not accept a request body.

1.18.2.2. Response

Example 1.125. 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	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}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

1.18.3.2. Response

Example 1.126. 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"
    }
  ]
}
```

This operation does not return a response body.

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}	String	The unique identifier of the tenant or account.

Example 1.127. 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"
  }
}
```

This operation does not accept a request body.

1.18.4.2. Response

Example 1.128. 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}	String	The unique identifier of the tenant or account.
{consistencygroup_id}	String	The unique identifier of the consistency group.

This operation does not accept a request body.

1.18.5.2. Response

Example 1.129. 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"
  }
}
```

This operation does not return a response body.

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}	String	The unique identifier of the tenant or account.
{consistencygroup_id}	String	The unique identifier of the consistency group.

Example 1.130. Delete consistency group: JSON request

```
{
    "consistencygroup": {
        "force": "False"
    }
}
```

This operation does not accept a request body.

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}	String	The unique identifier of the tenant or account.
{consistencygroup_id}	String	The unique identifier of the consistency group.

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

Name	Type	Description
name	String <i>(Optional)</i>	The consistency group name.
description	String <i>(Optional)</i>	The consistency group description.
add_volumes	String <i>(Optional)</i>	The UUID volume list for adding the volume consistency group.
remove_volumes	String <i>(Optional)</i>	The UUID volume list for removing the volume consistency group.

Example 1.131. 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

1.19.1.2. Response

Example 1.132. 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"
    }
  ]
}
```

This operation does not return a response body.

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 tenant ID in a multi-tenancy cloud.

Example 1.133. 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 1.134. 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

1.19.3.2. Response

Example 1.135. 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-c7ba3d18dded",
      "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}/cgssnapshots/{cgssnapshot_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 tenant ID in a multi-tenancy cloud.
{cgssnapshot_id}	String	The unique identifier of the consistency group snapshot.

This operation does not accept a request body.

1.19.4.2. Response

Example 1.136. Show consistency group snapshot details: JSON response

```
{
  "cgssnapshot": {
    "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 tenant ID in a multi-tenancy cloud.
{cgsnapshot_id}	String	The unique identifier 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 be attached 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 has occurred with the volume.
- ERROR_DELETING. There was an error deleting the volume.

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

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/defaults	Gets 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

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: 200, 300

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"
  }
}
```

Example 2.3. 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"
      }
    ]
  }
}
```

```
        "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_volid` 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 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.

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 unique identifier of the tenant or account.

Example 2.4. 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 2.5. 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 2.6. 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 2.7. 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 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 unique identifier of the tenant or account.

This operation does not accept a request body.

2.2.2.2. Response

Example 2.8. 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 2.9. 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 unique identifier of the tenant or account.

This operation does not accept a request body.

2.2.3.2. Response

Example 2.10. 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 2.11. 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 unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

This operation does not accept a request body.

2.2.4.2. Response

Example 2.12. 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 2.13. 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 unique identifier of the tenant or account.
{volume_id}	UUID	The unique identifier of an existing volume.

This operation does not accept a request body.

2.3. Volume types

Method	URI	Description
GET	/v1/{tenant_id}/types	Lists volume types.
POST	/v1/{tenant_id}/types	Creates 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 unique identifier of the tenant or account.

This operation does not accept a request body.

2.3.1.2. Response

Example 2.14. 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 2.15. 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 unique identifier of the tenant or account.

Example 2.16. 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 2.17. 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 2.18. 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 2.19. 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. 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

2.3.3.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{volume_type_id}	UUID	The unique identifier of an existing volume type.

This operation does not accept a request body.

2.3.3.2. Response

Example 2.20. 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 2.21. 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.

2.3.4. Delete volume type

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

Normal response codes: 202

2.3.4.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{volume_type_id}	UUID	The unique identifier of an existing volume type.

This operation does not accept a request body.

2.4. Snapshots

Method	URI	Description
POST	/v1/{tenant_id}/snapshots	Creates a snapshot.
GET	/v1/{tenant_id}/snapshots	Lists simple 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 unique identifier of the tenant or account.

Example 2.22. 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 2.23. 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 2.24. 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 2.25. 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 simple 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 unique identifier of the tenant or account.

This operation does not accept a request body.

2.4.2.2. Response

Example 2.26. 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 2.27. 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 unique identifier of the tenant or account.

This operation does not accept a request body.

2.4.3.2. Response

Example 2.28. 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 2.29. 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 unique identifier of the tenant or account.
{snapshot_id}	UUID	The unique identifier of an existing snapshot.

This operation does not accept a request body.

2.4.4.2. Response

Example 2.30. 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 2.31. 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 unique identifier of the tenant or account.
{snapshot_id}	UUID	The unique identifier of an existing 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 unique identifier of the tenant or account.
{snapshot_id}	UUID	The unique identifier of an existing snapshot.

This operation does not accept a request body.

2.4.6.2. Response

Example 2.32. 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 2.33. 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 unique identifier of the tenant or account.
{snapshot_id}	UUID	The unique identifier of an existing snapshot.

Example 2.34. Update snapshot metadata: XML request

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

Example 2.35. Update snapshot metadata: JSON request

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

This operation does not accept a request body.

2.4.7.2. Response

Example 2.36. 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 2.37. 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/defaults	Gets 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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID 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 admin tenant.

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

Name	Type	Description
usage	Boolean <i>(Optional)</i>	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 2.38. 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	String <i>(Required)</i>	A quota_set object.
cores	Int	The number of instance cores allowed for each tenant.

Name	Type	Description
	(Required)	
fixed_ips	Int (Required)	The number of fixed IP addresses 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 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 allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes allowed for each injected file path.
injected_files	Int (Required)	The number of injected files allowed for each tenant.
instances	Int (Required)	The number of instances allowed for each tenant.
key_pairs	Int (Required)	The number of key pairs allowed for each user.
metadata_items	Int (Required)	The number of metadata items allowed for each instance.
ram	Int (Required)	The amount of instance RAM in megabytes allowed for each tenant.
security_group_rules	Int (Optional)	The number of rules allowed for each security group.
security_groups	Int (Required)	The number of security groups 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 2.39. 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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID 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 admin tenant.

Example 2.40. 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	String <i>(Required)</i>	A quota_set object.
cores	Int <i>(Optional)</i>	The number of instance cores allowed for each tenant.
fixed_ips	Int <i>(Optional)</i>	The number of fixed IP addresses allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int <i>(Optional)</i>	The number of floating IP addresses allowed for each tenant.
id	Int <i>(Optional)</i>	The ID for the quota set.
injected_file_content_bytes	Int <i>(Optional)</i>	The number of bytes of content allowed for each injected file.
injected_file_path_bytes	Int <i>(Optional)</i>	The number of bytes allowed for each injected file path.
injected_files	Int <i>(Optional)</i>	The number of injected files allowed for each tenant.
instances	Int <i>(Optional)</i>	The number of instances allowed for each tenant.

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

Example 2.41. 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 2.42. 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	String <i>(Required)</i>	A quota_set object.
cores	Int <i>(Required)</i>	The number of instance cores allowed for each tenant.
fixed_ips	Int <i>(Required)</i>	The number of fixed IP addresses allowed for each tenant. Must be equal to or greater than the number of allowed instances.

Name	Type	Description
floating_ips	Int <i>(Required)</i>	The number of floating IP addresses 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 allowed for each injected file.
injected_file_path_bytes	Int <i>(Required)</i>	The number of bytes allowed for each injected file path.
injected_files	Int <i>(Required)</i>	The number of injected files allowed for each tenant.
instances	Int <i>(Required)</i>	The number of instances allowed for each tenant.
key_pairs	Int <i>(Required)</i>	The number of key pairs allowed for each user.
metadata_items	Int <i>(Required)</i>	The number of metadata items allowed for each instance.
ram	Int <i>(Required)</i>	The amount of instance RAM in megabytes allowed for each tenant.
security_group_rules	Int <i>(Optional)</i>	The number of rules allowed for each security group.
security_groups	Int <i>(Required)</i>	The number of security groups allowed for each tenant.

Example 2.43. 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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID 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 admin tenant.

This operation does not accept a request body.

2.5.4. Get default quotas

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

Normal response codes: 200

2.5.4.1. Request

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

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

This operation does not accept a request body.

2.5.4.2. Response

Example 2.44. Get default 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 get default quotas response:

Name	Type	Description
quota_set	String <i>(Required)</i>	A quota_set object.
cores	Int <i>(Required)</i>	The number of instance cores allowed for each tenant.
fixed_ips	Int <i>(Required)</i>	The number of fixed IP addresses 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 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 allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes allowed for each injected file path.
injected_files	Int (Required)	The number of injected files allowed for each tenant.
instances	Int (Required)	The number of instances allowed for each tenant.
key_pairs	Int (Required)	The number of key pairs allowed for each user.
metadata_items	Int (Required)	The number of metadata items allowed for each instance.
ram	Int (Required)	The amount of instance RAM in megabytes allowed for each tenant.
security_group_rules	Int (Optional)	The number of rules allowed for each security group.
security_groups	Int (Required)	The number of security groups allowed for each tenant.

Example 2.45. Get default 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.

2.5.5. 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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID of the tenant for which you want to show or update quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the admin tenant.
{user_id}	String	The user ID. Specify in the URI as a query string: user_id={user_id}.

This operation does not accept a request body.

2.5.5.2. Response

Example 2.46. 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	String <i>(Required)</i>	A quota_set object.
cores	Int <i>(Required)</i>	The number of instance cores allowed for each tenant.
fixed_ips	Int	The number of fixed IP addresses allowed for each tenant. Must be equal to or greater than the number of allowed instances.

Name	Type	Description
	(Required)	
floating_ips	Int	The number of floating IP addresses allowed for each tenant.
	(Required)	
id	Int	The ID for the quota set.
	(Required)	
injected_file_content_bytes	Int	The number of bytes of content allowed for each injected file.
	(Required)	
injected_file_path_bytes	Int	The number of bytes allowed for each injected file path.
	(Required)	
injected_files	Int	The number of injected files allowed for each tenant.
	(Required)	
instances	Int	The number of instances allowed for each tenant.
	(Required)	
key_pairs	Int	The number of key pairs allowed for each user.
	(Required)	
metadata_items	Int	The number of metadata items allowed for each instance.
	(Required)	
ram	Int	The amount of instance RAM in megabytes allowed for each tenant.
	(Required)	
security_group_rules	Int	The number of rules allowed for each security group.
	(Optional)	
security_groups	Int	The number of security groups allowed for each tenant.
	(Required)	

Example 2.47. 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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID of the tenant for which you want to show or update quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the admin tenant.
{user_id}	String	The user ID. Specify in the URI as a query string: user_id={user_id}.

Example 2.48. 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	String <i>(Required)</i>	A quota_set object.
cores	Int <i>(Optional)</i>	The number of instance cores allowed for each tenant.
fixed_ips	Int <i>(Optional)</i>	The number of fixed IP addresses allowed for each tenant. Must be equal to or greater than the number of allowed instances.
floating_ips	Int <i>(Optional)</i>	The number of floating IP addresses allowed for each tenant.
id	Int <i>(Optional)</i>	The ID for the quota set.
injected_file_content_bytes	Int <i>(Optional)</i>	The number of bytes of content allowed for each injected file.
injected_file_path_bytes	Int <i>(Optional)</i>	The number of bytes allowed for each injected file path.
injected_files	Int <i>(Optional)</i>	The number of injected files allowed for each tenant.

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

Example 2.49. 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 2.50. 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 <i>(Required)</i>	String	A quota_set object.
cores	Int	The number of instance cores allowed for each tenant.

Name	Type	Description
	(Required)	
fixed_ips	Int (Required)	The number of fixed IP addresses 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 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 allowed for each injected file.
injected_file_path_bytes	Int (Required)	The number of bytes allowed for each injected file path.
injected_files	Int (Required)	The number of injected files allowed for each tenant.
instances	Int (Required)	The number of instances allowed for each tenant.
key_pairs	Int (Required)	The number of key pairs allowed for each user.
metadata_items	Int (Required)	The number of metadata items allowed for each instance.
ram	Int (Required)	The amount of instance RAM in megabytes allowed for each tenant.
security_group_rules	Int (Optional)	The number of rules allowed for each security group.
security_groups	Int (Required)	The number of security groups allowed for each tenant.

Example 2.51. 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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID of the tenant for which you want to show or update quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the admin tenant.
{user_id}	String	The user ID. Specify in the URI as a query string: 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 tenant ID in a multi-tenancy cloud.
{tenant_id}	String	The ID of the tenant for which you want to show or update quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the admin tenant.
{user_id}	String	The user ID. Specify in the URI as a query string: user_id={user_id}.

This operation does not accept a request body.

2.5.8.2. Response

Example 2.52. 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": 10,
            "reserved": 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. Compute API (CURRENT)



Note

Effective in the OpenStack Kilo release, XML support in requests and responses was removed for the Compute API.

This documentation supports the API v2.1 base version. If all extensions are enabled, the Compute API v1.1 and v2.0 requests and responses match the API v2.1 requests and responses.

Each API version lists its required extensions for the full request and responses to be available. API v2.1 must enable all extensions all the time. It uses micro-version headers to expose any additional functionality.

This page lists the Compute API operations in the following order:

- [Versions](#)
- [Servers](#)
- [Flavors](#)
- [Keypairs](#)
- [Limits](#)
- [Extensions](#) in alphabetical order by resource

Method	URI	Description
API versions		
GET	/	Lists information about all Compute API versions.
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 details for all servers.
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 actions (servers, action)		
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a fixed IP address to a network on a server instance.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Attaches a volume to 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	Gets console output for a server instance.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a 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 a fixed 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	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	Unrescues a server. Changes status to ACTIVE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unshelves, or restores, a shelved server.
Servers admin actions (servers, action)		
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	Creates a back up of a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Injects network information into a server.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Locks 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	Pauses a server. Changes its status to PAUSED.
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.
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	Suspends a server and changes its status to SUSPENDED.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unlocks a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unpauses a paused server and changes its status to ACTIVE.
Servers diagnostics (servers, diagnostics)		
GET	/v2.1/{tenant_id}/servers/{server_id}/diagnostics	Gets 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 action (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 an action and server.
Servers password (servers, os-server-password)		
GET	/v2.1/{tenant_id}/servers/{server_id}/os-server-password	Gets the administrative password for a server.
Servers virtual interfaces (servers, os-virtual-interfaces)		

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Lists the virtual interfaces for an instance.
Flavors with extended attributes (flavors)		
GET	/v2.1/{tenant_id}/flavors	Lists flavors.
GET	/v2.1/{tenant_id}/flavors/{flavor_id}	Shows details for a flavor.
GET	/v2.1/{tenant_id}/flavors/detail	Lists flavors with details.
Flavors access (flavors, os-flavor-access, action)		
POST	/v2.1/{tenant_id}/flavors/os-flavor-access	Creates a flavor with access list.
GET	/v2.1/{tenant_id}/flavors/os-flavor-access	Lists flavors with access list information.
GET	/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}	Shows details for a flavor. Includes access list 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.
GET	/v2.1/{tenant_id}/flavors/os-flavor-access/detail	Lists flavors with details. Includes access list information.
Flavors extra-specs (flavors, os-flavor-extra-specs)		
POST	/v2.1/{tenant_id}/flavors/os-flavor-extra-specs/{flavor_id}	Creates and updates flavor extra specs.
GET	/v2.1/{tenant_id}/flavors/os-flavor-extra-specs/{flavor_id}	Lists extra specs for a flavor.
GET	/v2.1/{tenant_id}/flavors/os-flavor-extra-specs/{flavor_id}/{flavor_extra_spec_key}	Shows an extra spec for a flavor, by key.
PUT	/v2.1/{tenant_id}/flavors/os-flavor-extra-specs/{flavor_id}/{flavor_extra_spec_key}	Updates an extra spec value, by key, for a flavor.
Flavors manage (flavors, os-flavor-manage)		
POST	/v2.1/{tenant_id}/flavors/os-flavor-manage	Creates a flavor.
DELETE	/v2.1/{tenant_id}/flavors/os-flavor-manage/{flavor_id}	Deletes a flavor.
Keypairs (keypairs)		
GET	/v2.1/{tenant_id}/os-keypairs{?user_id}	Lists keypairs that are associated with the account.
POST	/v2.1/{tenant_id}/os-keypairs{?user_id}	Generates or imports a keypair.
DELETE	/v2.1/{tenant_id}/os-keypairs/{keypair_name}{?user_id}	Deletes a keypair.
GET	/v2.1/{tenant_id}/os-keypairs/{keypair_name}{?user_id}	Shows a keypair that is associated with the account.
Limits (limits)		
GET	/v2.1/{tenant_id}/limits	Shows global and rate limit information.
Extensions (extensions)		
GET	/v2.1/extensions	Lists available extensions.
GET	/v2.1/extensions/{alias}	Shows details for an extension.
Images		

Method	URI	Description
GET	/v2.1/images{?changes-since,server,name,status,type,limit,marker}	Lists IDs, names, and links for available images.
GET	/v2.1/images/detail{?changes-since,server,name,status,type,limit,marker}	Lists all details for available images.
GET	/v2.1/images/{image_id}	Gets details for an image.
DELETE	/v2.1/images/{image_id}	Deletes an image.
Image metadata		
GET	/v2.1/images/{image_id}/metadata	Shows metadata for an image.
PUT	/v2.1/images/{image_id}/metadata	Creates or replaces metadata for an image.
POST	/v2.1/images/{image_id}/metadata	Updates metadata items, by key, for an image.
GET	/v2.1/images/{image_id}/metadata/{key}	Shows details for a metadata item, by key, for an image.
PUT	/v2.1/images/{image_id}/metadata/{key}	Creates or updates a metadata item, by key, for an image.
DELETE	/v2.1/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/{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 aggregates id, name, and availability_zone for an aggregate.
GET	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Shows the details of an aggregate, hosts and metadata included.
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.
Attach Interfaces (os-attach-interfaces)		
POST	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces	Creates a port interface and uses it to attach a port to a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces	Lists port interfaces that are attached to a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces/{attachment_id}	Shows details for a port interface that is attached to a server.

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces/{attachment_id}	Detaches a port interface.
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.
Servers with block device mapping format (servers, os-block-device-mapping)		
POST	/v2.1/{tenant_id}/servers	Creates a server with a block device mapping.
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)		
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-cloudpipe/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}/os-consoles	Creates a console for a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-consoles	Lists all consoles for a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-consoles/{console_id}	Shows details for a console for a server instance.

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-soles/{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	Allocates a new floating IP address to a tenant or account.
DELETE	/v2.1/{tenant_id}/os-floating-ips/{id}	Deallocates a floating IP address.
Floating IPs bulk (os-floating-ips-bulk)		
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.
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/{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.

Method	URI	Description
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 that are available to the project.
POST	/v2.1/{tenant_id}/os-networks/add	Adds a network to a project.
GET	/v2.1/{tenant_id}/os-networks/{id}	Shows details for a network.
DELETE	/v2.1/{tenant_id}/os-networks/{id}	Deletes a network.
POST	/v2.1/{tenant_id}/os-networks/{id}/action	Associates a network with a host.
POST	/v2.1/{tenant_id}/os-networks/{id}/action	Disassociates the host from a network.
POST	/v2.1/{tenant_id}/os-networks/{id}/action	Disassociates a network from a project so that the network can be reused.
POST	/v2.1/{tenant_id}/os-networks/{id}/action	Disassociates the 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)		
DELETE	/v2.1/{tenant_id}/os-quota-sets	Deletes a quota for tenant.
PUT	/v2.1/{tenant_id}/os-quota-sets	Force-updates quota for tenant.
PUT	/v2.1/{tenant_id}/os-quota-sets	Updates quota for tenant.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-quota-sets/defaults	Shows default quotas for tenant.
GET	/v2.1/{tenant_id}/os-quota-sets/detail	Lists quotas with details for a tenant.
PUT	/v2.1/{tenant_id}/os-quota-sets/{?user_id}	Updates quota for user.
DELETE	/v2.1/{tenant_id}/os-quota-sets/{?user_id}	Deletes quota for 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.
Execute external events (os-server-external-events)		
POST	/v2.1/{tenant_id}/os-server-external-events	Creates events.
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/{ServerGroup_id}	Shows details for a server group.
DELETE	/v2.1/{tenant_id}/os-server-groups/{ServerGroup_id}	Deletes a server group.
Manage services (os-services)		
GET	/v2.1/{tenant_id}/os-services	Lists all services for a tenant. Includes reasons, if available, for why any disabled services were disabled.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-services/detail	Lists disabled services for a tenant. Includes reasons, if available, for why disabled services were disabled.
PUT	/v2.1/{tenant_id}/os-services/enable	Enables scheduling for a service.
PUT	/v2.1/{tenant_id}/os-services/disable	Disables scheduling for a service.
PUT	/v2.1/{tenant_id}/os-services/disable-log-reason	Logs information to the service table about why a service was disabled.
PUT	/v2.1/{tenant_id}/os-services/force-down	Forces down a service or unsets the forced-down state of a service.
DELETE	/v2.1/{tenant_id}/os-services	Deletes a service.
Usage reports (os-simple-tenant-usage)		
GET	/v2.1/os-simple-tenant-usage	Lists usage information for all tenants.
GET	/v2.1/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 project networks.
GET	/v2.1/{tenant_id}/os-tenant-networks/{id}	Shows details for a project network.
DELETE	/v2.1/{tenant_id}/os-tenant-networks/{id}	Deletes a project 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.
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.

3.1. API versions

Lists information for all API versions.

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

3.1.1. List API versions

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

Normal response codes: 200, 300

3.1.1.1. Request

This operation does not accept a request body.

3.1.1.2. Response

Example 3.1. 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": "2.12",
            "min_version": "2.1",
            "updated": "2013-07-23T11:33:21Z"
        }
    ]
}
```

3.2. Servers (servers)

Lists, creates, gets 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 is guaranteed to meet the security requirements set by the compute provider. For security reasons, the password is not returned in subsequent **GET** calls.

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 that can be supplied for each server. You can query this value through the `maxServerMeta` 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. This value can vary based on the image that is used to create the server.



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.

- You can inject only text files. You cannot inject binary or ZIP files into a new build.
- 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 is guaranteed to apply to all images in the deployment. Providers can set additional per-image personality limits.

The file injection might not occur until after the server is built and booted.

During file injection, any existing files that match files in the request are renamed to include the BAK extension appended with a time stamp. For example, if the /etc/passwd file exists, it is backed up as /etc/passwd.bak.1246036261.5785.

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, the addresses provided by the implementation cannot actually be used to access the server (from outside the local LAN). Here, a separate *access address* might be assigned 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. Nonetheless, clients that must access the server directly are encouraged to do so through an access address.

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 details for all servers.
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.

3.2.1. 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.
- 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, 203

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

3.2.1.1. Request

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

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

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

Name	Type	Description
changes-since <i>(Optional)</i>	DateTime	A time/date stamp for when the server last changed status.
image <i>(Optional)</i>	UUID	The UUID for the image.
flavor <i>(Optional)</i>	UUID	The UUID for the specific flavor, which is a combination of memory, disk size, and CPUs.
name <i>(Optional)</i>	Regexp	Name of the server as a string. Can be queried with regular expressions. The regular expression ?name=bob 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 <i>(Optional)</i>	ServerStatus	Value of the status of the server so that you can filter on "ACTIVE" for example.
host <i>(Optional)</i>	String	Name of the host as a string.
limit <i>(Optional)</i>	Int	Requests a page size of returned items from the query. 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 <i>(Optional)</i>	String	Specifies 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.

3.2.1.2. Response

Example 3.2. 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"
    }
]
```

3.2.2. 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, only the server ID, its links, and the admin password are guaranteed to be returned in the response. 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, ...), UnprocessableEntity (422), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), serverCapacityUnavailable (503)

3.2.2.1. Request

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

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

This list shows the body parameters for the request:

- **parameters:**

- **block_device_mapping_v2:**

- **device_name:** Csapi:string. Required.

Describes a path to the device for the volume that you want to use to boot the server.

- **source_type:** Csapi:string. Required.

The source type for the volume. A valid value is blank, snapshot, volume, or image.

- **destination_type:** Csapi:string. Optional.

Defines where the volume comes from. A valid value is local or volume.

- **delete_on_termination:** Csapi:string. Required.

To delete the boot volume when the server stops, specify true. Otherwise, specify false.

- **guest_format:** Csapi:string. Required.

Specifies the guest server disk file system format, such as ephemeral or swap.

- **boot_index:** Csapi:string. Required.

The boot order of the device. Use -1 for the boot volume. Use 0 for an attached volume.

Example 3.3. 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"
        }
    }
}
```

3.2.2.2. Response

Example 3.4. Create server: JSON response

```
{
    "server": {
        "OS-DCF:diskConfig": "AUTO",
        "adminPass": "zPnp2GseTqG4",
        "id": "8195065cfea4-4d57-b93f-5c5c63fe90e8",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/servers/8195065cfea4-4d57-b93f-5c5c63fe90e8",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/8195065cfea4-4d57-b93f-5c5c63fe90e8",
                "rel": "bookmark"
            }
        ],
        "security_groups": [
            {
                "name": "default"
            }
        ]
    }
}
```

3.2.3. List details for servers

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/detail {?changes-since,image,flavor,name, status,host,limit,marker}	Lists details for all servers.

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 specified 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, 203

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

3.2.3.1. Request

This table shows the URI parameters for the list details for servers request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID 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)	A time/date stamp for when the server last changed status.
image	UUID (Optional)	The UUID for the image.
flavor	UUID	The UUID for the specific flavor, which is a combination of memory, disk size, and CPUs.

Name	Type	Description
	(Optional)	
name	Regexp (Optional)	Name of the server as a string. Can be queried with regular expressions. The regular expression ?name=bob 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)	Value of the status of the server so that you can filter on "ACTIVE" for example.
host	String (Optional)	Name of the host as a string.
limit	Int (Optional)	Requests a page size of returned items from the query. 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)	Specifies 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.

3.2.3.2. Response

Example 3.5. 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": "9cbefc35-d372-40c5-88e2-9fdab6ea12c",
        "image": {
          "id": "70a599e0-31e7-49b7-b260-868f441e862b",
          "minDisk": 1,
          "minMemory": 256,
          "name": "Ubuntu 12.04 LTS (Precise Pangolin) - Public Image",
          "os_type": "Ubuntu",
          "size": 20
        }
      }
    }
  ]
}
```

```
        "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-d372-40c5-88e2-9fdalb6ea12c",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
9cbefc35-d372-40c5-88e2-9fdalb6ea12c",
                "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",
        "tenant_id": "openstack",
        "updated": "2013-10-31T06:32:32Z",
        "user_id": "fake"
    }
]
```

3.2.4. Get 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 specified 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, 203

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

3.2.4.1. Request

This table shows the URI parameters for the get server details request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.2.4.2. Response

Example 3.6. Get 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-4fb6-4elf-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-4fb6-4elf-b1d5-d6bf4437f7d5",
        "rel": "self"
    },
    {
        "href": "http://openstack.example.com/openstack/servers/f474386b-4fb6-4elf-b1d5-d6bf4437f7d5",
        "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": "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",
"tenant_id": "openstack",
"updated": "2013-10-31T07:31:30Z",
"user_id": "fake"
}
```

3.2.5. 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.

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)

3.2.5.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.7. Update Server Name Request: JSON

```
{
  "server": {
    "name": "new-server-test"
  }
}
```

Example 3.8. Update Server IP Addresses Request: JSON

```
{
  "server": {
    "accessIPv4": "67.23.10.132",
    "accessIPv6": "::babe:67.23.10.132"
  }
}
```

3.2.5.2. Response

Example 3.9. Update Server Name Response: JSON

```
{
  "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": "67.23.10.138",
    "accessIPv6": "::babe:67.23.10.138",
    "progress": 0,
    "status": "active"
  }
}
```

```
        "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": "67.23.10.138"
                },
                {
                    "version": 6,
                    "addr": "::babe:67.23.10.138"
                }
            ],
            "private": [
                {
                    "version": 4,
                    "addr": "10.176.42.19"
                },
                {
                    "version": 6,
                    "addr": "::babe:10.176.42.19"
                }
            ]
        },
        "links": [
            {

```

```

        "rel": "self",
        "href": "http://servers.api.openstack.org/v2/1234/servers/
52415800-8b69-11e0-9b19-734fcece0043"
    },
    {
        "rel": "bookmark",
        "href": "http://servers.api.openstack.org/1234/servers/
52415800-8b69-11e0-9b19-734fcece0043"
    }
]
}
}

```

Example 3.10. Update Server IP Addresses Response: JSON

```

{
    "server": {
        "id": "52415800-8b69-11e0-9b19-734f565bc83b",
        "tenant_id": "1234",
        "user_id": "5678",
        "name": "new-server-test",
        "created": "2010-11-11T12:00:00Z",
        "updated": "2010-11-12T12:55:55Z",
        "hostId": "e4d909c290d0fb1ca068ffaddf22cbd0",
        "accessIPv4": "67.23.10.132",
        "accessIPv6": "::babe:67.23.10.132",
        "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": "67.23.10.138"
            },
            {
                "version": 6,
                "addr": "::babe:67.23.10.138"
            }
        ],
        "private": [
            {
                "version": 4,
                "addr": "10.176.42.19"
            },
            {
                "version": 6,
                "addr": "::babe:10.176.42.19"
            }
        ]
    },
    "links": [
        {
            "rel": "self",
            "href": "http://servers.api.openstack.org/v2/1234/servers/
52415800-8b69-11e0-9b19-734fcece0043"
        },
        {
            "rel": "bookmark",
            "href": "http://servers.api.openstack.org/1234/servers/
52415800-8b69-11e0-9b19-734fcece0043"
        }
    ]
}
```

3.2.6. 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 server's status is not locked.
- You can delete a server which status is not locked.
- If the server is locked, you need to 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 is deleted from the list of servers returned by an API calls.
- The server managed by OpenStack Compute is deleted on the compute node.

Troubleshooting

- If server status remains in `deleting` or shows another error status, the request failed. Ensure you meet the preconditions then investigate the compute back end.
- The request returns the HTTP 409 status code when the server is locked even if you have a correct permissions. Ensure 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)

3.2.6.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.3. 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.

3.3.1. Create multiple servers

Method	URI	Description
POST	/v2.1/{tenant_id}/servers	Creates one or more servers.

Normal response codes: 202

3.3.1.1. Request

This table shows the URI parameters for the create multiple servers request:

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

This list shows the body parameters for the request:

- **parameters:**

- **block_device_mapping_v2:**

- **device_name:** Csapi:string. Required.

Describes a path to the device for the volume that you want to use to boot the server.

- **source_type:** Csapi:string. Required.

The source type for the volume. A valid value is blank, snapshot, volume, or image.

- **destination_type:** Csapi:string. Optional.

Defines where the volume comes from. A valid value is local or volume.

- **delete_on_termination:** Csapi:string. Required.

To delete the boot volume when the server stops, specify true. Otherwise, specify false.

- **guest_format:** Csapi:string. Required.

Specifies the guest server disk file system format, such as ephemeral or swap.

- **boot_index:** Csapi:string. Required.

The boot order of the device. Use -1 for the boot volume. Use 0 for an attached volume.

Example 3.11. 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"
    }
}
```

3.3.1.2. Response

Example 3.12. Create multiple servers without reservation ID

```
{
    "server": {
        "OS-DCF:diskConfig": "AUTO",
        "adminPass": "zPnp2GseTqG4",
        "id": "8195065cfea4-4d57-b93f-5c5c63fe90e8",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/servers/
8195065cfea4-4d57-b93f-5c5c63fe90e8",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
8195065cfea4-4d57-b93f-5c5c63fe90e8",
                "rel": "bookmark"
            }
        ],
        "security_groups": [
            {
                "name": "default"
            }
        ]
    }
}
```

3.3.2. 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

3.3.2.1. Request

This table shows the URI parameters for the create multiple servers with reservation id request:

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

This list shows the body parameters for the request:

- **parameters:**

- **block_device_mapping_v2:**

- **device_name:** Csapi:string. Required.

Describes a path to the device for the volume that you want to use to boot the server.

- **source_type:** Csapi:string. Required.

The source type for the volume. A valid value is blank, snapshot, volume, or image.

- **destination_type:** Csapi:string. Optional.

Defines where the volume comes from. A valid value is local or volume.

- **delete_on_termination:** Csapi:string. Required.

To delete the boot volume when the server stops, specify true. Otherwise, specify false.

- **guest_format:** Csapi:string. Required.

Specifies the guest server disk file system format, such as ephemeral or swap.

- **boot_index:** Csapi:string. Required.

The boot order of the device. Use -1 for the boot volume. Use 0 for an attached volume.

Example 3.13. 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"
  }
}
```

3.3.2.2. Response

Example 3.14. Create multiple servers with reservation ID

```
{
  "reservation_id": "r-3fhpjulh"
}
```

3.4. Servers actions (servers, action)

Performs actions on a server. Specify the action in the request body.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a fixed IP address to a network on a server instance.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Attaches a volume to 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	Gets console output for a server instance.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a 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.

Method	URI	Description
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 a fixed 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	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	Unrescues a server. Changes status to ACTIVE.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unshelves, or restores, a shelved server.

3.4.1. Add fixed IP to server (addFixedIp action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a fixed IP address to a network on a server instance.

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

3.4.1.1. Request

This table shows the URI parameters for the add fixed ip to server (addfixedip action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.15. Add fixed IP to server (addFixedIp action): JSON request

```
{
    "addFixedIp": {
        "networkId": 1
    }
}
```

3.4.2. Attach volume (attach action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Attaches a volume to a server.

Specify the attach action in the request body.

If the attach operation succeeds, the volume status is `in-use`.

Preconditions

- The server must exist.
- You can only attach a volume to the server when its status is `available`.
- The connection to the Block Storage service is valid.

Troubleshooting

- If the request fails due to an OpenStack Compute service error, ensure you meet the pre-conditions 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)

3.4.2.1. Request

This table shows the URI parameters for the attach volume (attach action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.16. Attach volume (attach action): JSON request

```
{
  "attach": {
    "volume_id": "15e59938-07d5-11e1-90e3-e3dffe0c5983",
    "device": "/dev/vdb",
    "disk_bus": "ide",
    "device_type": "cdrom"
  }
}
```

3.4.3. 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)

3.4.3.1. Request

This table shows the URI parameters for the confirm resized server (confirmresize action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.17. Confirm resized server (confirmResize action): JSON request

```
{
    "confirmResize": null
}
```

3.4.4. 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 `available` 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)

3.4.4.1. Request

This table shows the URI parameters for the create image (createimage action) request:

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

Name	Type	Description
{server_id}	UUID	The UUID for the server.

Example 3.18. Create image (createImage action): JSON request

```
{
    "createImage": {
        "name": "foo-image",
        "metadata": {
            "meta_var": "meta_val"
        }
    }
}
```

3.4.5. 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

3.4.5.1. Request

This table shows the URI parameters for the evacuate server (evacuate action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.19. Evacuate server (evacuate action): JSON request

```
{
  "evacuate": {
    "host": "b419863b7d814906a68fb31703c0dbd6",
    "admin_password": "MySecretPass",
    "on_shared_storage": "False"
  }
}
```

3.4.5.2. Response

Example 3.20. Evacuate server (evacuate action): JSON response

```
{
  "admin_password": "MySecretPass"
}
```

3.4.6. 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

3.4.6.1. Request

This table shows the URI parameters for the force-delete server (forcedelete action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.21. Force-delete server (forceDelete action): JSON request

```
{
    "forceDelete": null
}
```

3.4.7. Get console output for a server (os-getConsoleOutput action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets console output for a server instance.

Specify the os-getConsoleOutput action in the request body.

Normal response codes: 200

3.4.7.1. Request

This table shows the URI parameters for the get console output for a server (os-getconsoleoutput action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.22. Get console output: JSON request

```
{
    "os-getConsoleOutput": {
        "length": 50
    }
}
```

3.4.7.2. Response

Example 3.23. Get console output: JSON response

```
{
    "output": "FAKE CONSOLE OUTPUT\nANOTHER\nLAST LINE"
}
```

3.4.8. Get RDP console for a server (os-getRDPConsole action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Gets a RDP console for a server.

Specify the os-getRDPConsole action in the request body.

Normal response codes: 200

3.4.8.1. Request

This table shows the URI parameters for the get rdp console for a server (os-getrdpconsole action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.24. Get RDP console for a server (os-getRDPConsole action): JSON request

```
{
  "os-getRDPConsole": {
    "type": "rdp-html5"
  }
}
```

3.4.8.2. Response

Example 3.25. Get RDP console for a server (os-getRDPConsole action): JSON response

```
{
  "console": {
    "type": "rdp-html5",
    "url": "http://127.0.0.1:6083/?token=191996c3-7b0f-42f3-95a7-f1839f2da6ed"
  }
}
```

3.4.9. Get serial console for a server (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

3.4.9.1. Request

This table shows the URI parameters for the get serial console for a server (os-getserialconsole action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.26. Get serial console for a server (os-getSerialConsole action): JSON request

```
{
  "os-getSerialConsole": {
    "type": "serial"
  }
}
```

3.4.9.2. Response

Example 3.27. Get serial console for a server (os-getSerialConsole action): JSON response

```
{
  "console": {
    "type": "serial",
    "url": "ws://127.0.0.1:6083/?token=f9906a48-b71e-4f18-baca-c987da3ebdb3"
  }
}
```

3.4.10. Get SPICE console for a server (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

3.4.10.1. Request

This table shows the URI parameters for the get spice console for a server (os-getspiceconsole action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.28. Get SPICE console for a server (os-getSPICEConsole action): JSON request

```
{
  "os-getSPICEConsole": {
    "type": "spice-html5"
  }
}
```

3.4.10.2. Response

Example 3.29. Get SPICE console for a server (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"
  }
}
```

3.4.11. Get VNC console for a server (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

3.4.11.1. Request

This table shows the URI parameters for the get vnc console for a server (os-getvncconsole action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.30. Get VNC console for a server (os-getVNCConsole action): JSON request

```
{
  "os-getVNCConsole": {
    "type": "novnc"
  }
}
```

3.4.11.2. Response

Example 3.31. Get VNC console for a server (os-getVNCConsole action): JSON response

```
{
  "console": {
    "type": "novnc",
    "url": "http://127.0.0.1:6080/vnc_auto.html?token=191996c3-7b0f-42f3-95a7-f1839f2da6ed"
  }
}
```

3.4.12. 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)

3.4.12.1. Request

This table shows the URI parameters for the reboot server (reboot action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.32. Reboot server (reboot action): JSON request

```
{
    "reboot": {
        "type": "HARD"
    }
}
```

3.4.13. 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)

3.4.13.1. Request

This table shows the URI parameters for the rebuild server (rebuild action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.33. 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": "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcYBpbBqdXN0IHN1Y2ggYSBkaXJ1Y3Rpb24gYW5kIGF0IHN1Y2ggYSBzcGV1ZC4uLk10IGZlZWxzIGFuIGltcHVs
c21vb4uLnRoaXMgaXMgdGh1IHBsYWN1IHRvIGdvIG5vdy4gQnV0IHRoZSBza3kga25vd3MgdGh1IHZ1YXNvbnMgYW5kIHRoZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3VkcwgyYW5kIHLvdSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlmdCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcm16b25zLiINCg0KLVJpy2hhcmQgQmFjaA=="
            }
        ],
        "preserve_ephemeral": true
    }
}
```

{

3.4.13.2. Response

Example 3.34. Rebuild server (rebuild action): JSON response

```
{  
    "server": {  
        "OS-DCF:diskConfig": "AUTO",  
        "accessIPv4": "1.2.3.4",  
        "accessIPv6": "80fe:::",  
        "addresses": {  
            "private": [  
                {  
                    "addr": "192.168.0.3",  
                    "version": 4  
                }  
            ]  
        },  
        "adminPass": "seekr3t",  
        "created": "2012-09-25T13:36:08Z",  
        "flavor": {  
            "id": "1",  
            "links": [  
                {  
                    "href": "http://openstack.example.com/openstack/flavors/  
1",  
                    "rel": "bookmark"  
                }  
            ]  
        },  
        "hostId": "935dc1019fd43814a1d2a6e9b320dcac352d3a02c69f8be7ba41002",  
        "id": "27568e59-cfb7-4283-a00e-4af933f2d539",  
        "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/openstack/servers/  
27568e59-cfb7-4283-a00e-4af933f2d539",  
                "rel": "self"  
            },  
            {  
                "href": "http://openstack.example.com/openstack/servers/  
27568e59-cfb7-4283-a00e-4af933f2d539",  
                "rel": "bookmark"  
            }  
        ],  
        "metadata": {  
            "meta var": "meta val"  
        },  
        "name": "foobar",  
        "progress": 0,  
    }  
}
```

```
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2012-09-25T13:36:09Z",
        "user_id": "fake"
    }
}
```

3.4.14. Remove fixed IP from server (removeFixedIp action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Removes 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

3.4.14.1. Request

This table shows the URI parameters for the remove fixed ip from server (removefixedip action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.35. Remove fixed IP from server (removeFixedIp action): JSON request

```
{
    "removeFixedIp": {
        "address": "10.0.0.4"
    }
}
```

3.4.15. 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 UNRESCUE.

Normal response codes: 202

3.4.15.1. Request

This table shows the URI parameters for the rescue server (rescue action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.36. Rescue server: JSON request

```
{
  "rescue": {
    "adminPass": "MySecretPass"
  }
}
```

Example 3.37. Rescue server with image: JSON request

```
{
  "rescue": {
    "adminPass": "MySecretPass",
    "rescue_image_ref": "70a599e0-31e7-49b7-b260-868f441e862b"
  }
}
```

3.4.15.2. Response

Example 3.38. Extended rescue server: JSON response

```
{
  "adminPass": "MySecretPass"
}
```

3.4.16. 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 auto_confirm option of the Compute service to True, the Compute service automatically confirms the resize operation.

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)

3.4.16.1. Request

This table shows the URI parameters for the resize server (resize action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.39. Resize server (resize action): JSON request

```
{
    "resize": {
        "flavorRef": "2"
    }
}
```

3.4.17. 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

3.4.17.1. Request

This table shows the URI parameters for the restore soft-deleted instance (restore action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.40. Restore soft-deleted instance (restore action): JSON request

```
{  
    "restore": null  
}
```

3.4.18. 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)

3.4.18.1. Request

This table shows the URI parameters for the revert resized server (revertresize action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.41. Revert resized server (revertResize action): JSON request

```
{
    "revertResize": null
}
```

3.4.19. Shelve 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

3.4.19.1. Request

This table shows the URI parameters for the shelve server (shelve action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.42. Shelve server: JSON request

```
{
```

```
        "shelve": null  
    }
```

3.4.20. 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

3.4.20.1. Request

This table shows the URI parameters for the shelf-offload (remove) server (shelveoffload action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.43. Shelve server: JSON request

```
{
    "shelveOffload": null
}
```

3.4.21. 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

3.4.21.1. Request

This table shows the URI parameters for the start server (os-start action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.44. Start server: JSON request

```
{
    "os-start": null
}
```

3.4.22. 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

3.4.22.1. Request

This table shows the URI parameters for the stop server (os-stop action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.45. Stop server: JSON request

```
{
    "os-stop": null
}
```

3.4.23. 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

3.4.23.1. Request

This table shows the URI parameters for the unrescue server (unrescue action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.46. Unrescue server: JSON request

```
{
    "unrescue": null
}
```

3.4.24. 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

3.4.24.1. Request

This table shows the URI parameters for the unshelve (restore) shelved server (unshelve action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.47. Unshelve server: JSON request

```
{
    "unshelve": null
}
```

3.5. Servers admin actions (servers, action)

Administrators only. Performs actions on a server. Specify the action in the request body.

Method	URI	Description
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	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	Locks 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	Pauses a server. Changes its status to PAUSED.
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.
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	Suspends a server and changes its status to SUSPENDED.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unlocks a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unpauses a paused server and changes its status to ACTIVE.

3.5.1. 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

3.5.1.1. Request

This table shows the URI parameters for the change administrative password (changePassword action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.48. Change administrative password (changePassword action): JSON request

```
{
    "changePassword": {
        "adminPass": "foo"
    }
}
```

3.5.2. 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

3.5.2.1. Request

This table shows the URI parameters for the create server back up (createbackup action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.49. Create server back up (createBackup action): JSON request

```
{
    "createBackup": {
        "name": "Backup 1",
        "backup_type": "daily",
        "rotation": 1
    }
}
```

3.5.3. 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

3.5.3.1. Request

This table shows the URI parameters for the inject network information (injectnetworkinfo action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.50. Inject network information (injectNetworkInfo action): JSON request

```
{
    "injectNetworkInfo": null
}
```

3.5.4. 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

3.5.4.1. Request

This table shows the URI parameters for the lock server (lock action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.51. Lock server (lock action): JSON request

```
{  
    "lock": null  
}
```

3.5.5. 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

3.5.5.1. Request

This table shows the URI parameters for the migrate server (migrate action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.52. Migrate server (migrate action): JSON request

```
{  
    "migrate": null  
}
```

3.5.6. 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.

Specify the os-migrateLive 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

3.5.6.1. Request

This table shows the URI parameters for the live-migrate server (os-migrateLive action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.53. Live-migrate server (os-migrateLive action): JSON request

```
{
    "os-migrateLive": {
        "host": "01c0cadef72d47e28a672a76060d492c",
        "block_migration": false,
        "disk_over_commit": false
    }
}
```

3.5.7. 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

3.5.7.1. Request

This table shows the URI parameters for the pause server (pause action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.54. Pause server (pause action): JSON request

```
{  
    "pause": null  
}
```

3.5.8. 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

3.5.8.1. Request

This table shows the URI parameters for the reset networking on a server (resetnetwork action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.55. Reset networking on a server (resetNetwork action): JSON request

```
{
    "resetNetwork": null
}
```

3.5.9. 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

3.5.9.1. Request

This table shows the URI parameters for the reset server state (os-resetstate action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.56. Reset server state (os-resetState action): JSON request

```
{
  "os-resetState": {
    "state": "active"
  }
}
```

3.5.10. 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

3.5.10.1. Request

This table shows the URI parameters for the resume suspended server (resume action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.57. Resume suspended server (resume action): JSON request

```
{  
    "resume": null  
}
```

3.5.11. Suspend server (suspend action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Suspends a server and changes its status to SUSPENDED.

Specify the suspend 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

3.5.11.1. Request

This table shows the URI parameters for the suspend server (suspend action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.58. Suspend server (suspend action): JSON request

```
{  
    "suspend": null  
}
```

3.5.12. Unlock server (unlock action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Unlocks a 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

3.5.12.1. Request

This table shows the URI parameters for the unlock server (unlock action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.59. Unlock server (unlock action): JSON request

```
{  
    "unlock": null  
}
```

3.5.13. 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

3.5.13.1. Request

This table shows the URI parameters for the unpause server (unpause action) request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.60. Unpause server (unpause action): JSON request

```
{
    "unpause": null
}
```

3.6. Servers diagnostics (servers, diagnostics)

Gets the usage data for a server.

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/diagnostics	Gets basic usage data for a server.

3.6.1. Get server diagnostics

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/diagnostics	Gets 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

3.6.1.1. Request

This table shows the URI parameters for the get server diagnostics request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.6.1.2. Response

Example 3.61. Server diagnostics: JSON response

```
{
    "cpu0_time": 17300000000,
    "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
}
```

3.7. Servers IPs (servers, ips)

Lists the IP addresses assigned to an instance or 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.

3.7.1. 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

3.7.1.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.7.1.2. Response

Example 3.62. List IPs: JSON response

```
{
    "addresses": {
        "private": [
            {
                "addr": "192.168.0.3",
                "version": 4
            }
        ]
    }
}
```

3.7.2. 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

3.7.2.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.
{network_label}	String	The network label, such as <code>public</code> or <code>private</code> .

This operation does not accept a request body.

3.7.2.2. Response

Example 3.63. Show IP details: JSON response

```
{
  "private": [
    {
      "addr": "192.168.0.3",
      "version": 4
    }
  ]
}
```

3.8. 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.

3.8.1. 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: 200203

3.8.1.1. Request

This table shows the URI parameters for the list all metadata request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.8.1.2. Response

Example 3.64. List all metadata: JSON response

```
{
  "metadata": {
    "foo": "Foo Value"
  }
}
```

Example 3.65. List all metadata: JSON response

```
{
  "metadata": {
    "foo": "Foo Value"
  }
}
```

3.8.2. 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.

If this operation exceeds the quota for metadata items, the API throws an `overLimit` (413) fault.

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

3.8.2.1. Request

This table shows the URI parameters for the create or replace metadata items request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.66. Create or replace metadata items: JSON request

```
{
  "metadata": {
    "foo": "Foo Value"
  }
}
```

3.8.2.2. Response

Example 3.67. Create or replace metadata items: JSON response

```
{
  "metadata": {
    "foo": "Foo Value"
  }
}
```

3.8.3. 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 the specified keys. Does not modify items that are not specified in the request.

If this operation exceeds the quota for metadata items, the API throws an `overLimit` (413) fault.

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

3.8.3.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

Example 3.68. Update metadata items: JSON request

```
{
  "metadata": {
    "foo": "Foo Value"
  }
}
```

3.8.3.2. Response

Example 3.69. Update metadata items: JSON response

```
{
  "metadata": {
    "foo": "Foo Value"
  }
}
```

3.8.4. 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: 200203

3.8.4.1. Request

This table shows the URI parameters for the show metadata item details request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.
{key}	String	The metadata key.

This operation does not accept a request body.

3.8.4.2. Response

Example 3.70. Show metadata item details: JSON response

```
{
  "meta": {
    "foo": "Foo Value"
  }
}
```

Example 3.71. Show metadata item details: JSON response

```
{
  "meta": {
    "foo": "Foo Value"
  }
}
```

3.8.5. 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.

If this operation exceeds the quota for metadata items, the API throws an `overLimit` (413) fault.

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

3.8.5.1. Request

This table shows the URI parameters for the create or update metadata item request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.
{key}	String	The metadata key.

Example 3.72. Create or update metadata item: JSON request

```
{
  "meta": {
    "foo": "Bar Value"
  }
}
```

3.8.5.2. Response

Example 3.73. Create or update metadata item: JSON response

```
{
  "meta": {
    "foo": "Bar Value"
  }
}
```

3.8.6. 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

3.8.6.1. Request

This table shows the URI parameters for the delete metadata item request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.
{key}	String	The metadata key.

This operation does not accept a request body.

3.9. Servers action (servers, os-instance-actions)

Permits all users to list available server actions for a server. Permits administrators to get details for a server action for a server.

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 an action and server.

3.9.1. 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

3.9.1.1. Request

This table shows the URI parameters for the list actions for server request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.9.1.2. Response

Example 3.74. List actions for server: JSON response

```
{
  "instanceActions": [
    {
      "action": "resize",
      "server_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",
      "server_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"
    }
  ]
}
```

3.9.2. Show server action details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-instance-actions/{request_id}	Shows details for an action and server.

Normal response codes: 200

3.9.2.1. Request

This table shows the URI parameters for the show server action details request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.9.2.2. Response

Example 3.75. 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": ""
      }
    ],
    "server_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"
  }
}
```

3.10. Servers password (servers, os-server-password)

Gets the encrypted administrative password set through the metadata service.

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-server-password	Gets the administrative password for a server.

3.10.1. Get server password

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-server-password	Gets the administrative password 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

3.10.1.1. Request

This table shows the URI parameters for the get server password request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.10.1.2. Response

Example 3.76. Get server password: JSON response

```
{
  "password": "xlozO3wLCBRWAa2yDjCCVx8vwNPypxnypmRYDa/zErlQ+EzPe1S/
Gz6nfmC52m01OSCRUOmG7kqqgejPof6M7b0ezS387zjq4LSvvwp28zUknzy4YzfFGhnHAdai3TxUJ26pfQCYrq8UTzm
I1K2LsuipfxSJR7Wdke4zNXJjHHP2RfYsVbZ/k9ANu+Nz4iIH8/7Cacud/
pphH7EjrY6a4RZNrjQskrhKYed0YERpotyjYk1eDtRe72GrSiXteqCM4biaQ5w3ruS+AcX//"
  }
```

3.11. 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.

3.11.1. 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, but the API v2.12 micro-version returns it in the `net_id` attribute.

Normal response codes: 202

3.11.1.1. Request

This table shows the URI parameters for the list virtual interfaces request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.11.1.2. Response

Example 3.77. List virtual interfaces: JSON response

```
{
    "virtual_interfaces": [
        {
            "id": "cec8b9bb-5d22-4104-b3c8-4c35db3210a6",
            "mac_address": "fa:16:3e:3c:ce:6f"
        }
    ]
}
```

3.12. Flavors with extended attributes (flavors)

Shows information about flavors.

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors	Lists flavors.
GET	/v2.1/{tenant_id}/flavors/{flavor_id}	Shows details for a flavor.
GET	/v2.1/{tenant_id}/flavors/detail	Lists flavors with details.

3.12.1. List flavors

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors	Lists flavors.

Normal response codes: 200

3.12.1.1. Request

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

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

3.12.1.2. Response

Example 3.78. List flavors: JSON response

```
{
    "flavors": [
        {
            "id": "1",
            "links": [
                {
                    "href": "http://openstack.example.com/v2/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/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/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/openstack/
flavors/4",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/flavors/
4",
            "rel": "bookmark"
        }
    ],
    "name": "m1.large"
},
{
    "id": "5",
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/
flavors/5",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/flavors/
5",
            "rel": "bookmark"
        }
    ],
    "name": "m1.xlarge"
}
]
```

3.12.2. Show flavor details

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/{flavor_id}	Shows details for a flavor.

Normal response codes: 200

3.12.2.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{flavor_id}	UUID	The flavor ID.

3.12.2.2. Response

Example 3.79. Show flavor details: JSON response

```
{
  "flavor": {
    "OS-FLV-DISABLED:disabled": false,
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "disk": 1,
    "id": "1",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/flavors/1",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/1",
        "rel": "bookmark"
      }
    ],
    "name": "m1.tiny",
    "os-flavor-access:is_public": true,
    "ram": 512,
    "rxtx_factor": 1.0,
    "swap": "",
    "vcpus": 1
  }
}
```

3.12.3. List flavors with details

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/detail	Lists flavors with details.

Normal response codes: 200

3.12.3.1. Request

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

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

3.12.3.2. Response

Example 3.80. 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/openstack/
flavors/1",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/flavors/
1",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.tiny",
            "ram": 512,
            "rxtx_factor": 1.0,
            "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/openstack/
flavors/2",
                    "rel": "self"
                },
                {

```

```
                "href": "http://openstack.example.com/openstack/flavors/
2",
                "rel": "bookmark"
            },
        ],
        "name": "m1.small",
        "ram": 2048,
        "rxtx_factor": 1.0,
        "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/openstack/
flavors/3",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/flavors/
3",
                "rel": "bookmark"
            }
        ],
        "name": "m1.medium",
        "ram": 4096,
        "rxtx_factor": 1.0,
        "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/openstack/
flavors/4",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/flavors/
4",
                "rel": "bookmark"
            }
        ],
        "name": "m1.large",
        "ram": 8192,
        "rxtx_factor": 1.0,
        "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/openstack/
flavors/5",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/flavors/
5",
            "rel": "bookmark"
        }
    ],
    "name": "m1.xlarge",
    "ram": 16384,
    "rxtx_factor": 1.0,
    "swap": "",
    "vcpus": 8
}
]
}

```

3.13. Flavors access (flavors, os-flavor-access, action)

Provides flavor access support.

Method	URI	Description
POST	/v2.1/{tenant_id}/flavors/os-flavor-access	Creates a flavor with access list.
GET	/v2.1/{tenant_id}/flavors/os-flavor-access	Lists flavors with access list information.
GET	/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}	Shows details for a flavor. Includes access list 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.
GET	/v2.1/{tenant_id}/flavors/os-flavor-access/detail	Lists flavors with details. Includes access list information.

3.13.1. Create flavor with access list

Method	URI	Description
POST	/v2.1/{tenant_id}/flavors/os-flavor-access	Creates a flavor with access list.

Normal response codes: 200

3.13.1.1. Request

This table shows the URI parameters for the create flavor with access list request:

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

Example 3.81. Create flavor with access list: JSON request

```
{
  "flavor": {
    "name": "test_flavor",
    "ram": 1024,
    "vcpus": 2,
    "disk": 10,
    "id": "10",
    "os-flavor-access:is_public": false
  }
}
```

3.13.1.2. Response

Example 3.82. Create flavor with access list: JSON response

```
{
  "flavor": {
    "OS-FLV-DISABLED:disabled": false,
    "disk": 10,
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "os-flavor-access:is_public": false,
    "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
  }
}
```


3.13.2. List flavors with access list

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/os-flavor-access	Lists flavors with access list information.

Normal response codes: 200

3.13.2.1. Request

This table shows the URI parameters for the list flavors with access list request:

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

This operation does not accept a request body.

3.13.2.2. Response

Example 3.83. List flavors with access list: JSON response

```
{
    "flavor_access": [
        {
            "flavor_id": "10",
            "tenant_id": "fake_tenant"
        }
    ]
}
```

3.13.3. Show flavor details with access list

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}	Shows details for a flavor. Includes access list information.

Normal response codes: 200

3.13.3.1. Request

This table shows the URI parameters for the show flavor details with access list request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{flavor_id}	UUID	The flavor ID.

This operation does not accept a request body.

3.13.3.2. Response

Example 3.84. Show flavor details with access list: 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/flavors/1",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/flavors/1",
        "rel": "bookmark"
      }
    ],
    "name": "m1.tiny",
    "ram": 512,
    "swap": "",
    "vcpus": 1
  }
}
```

3.13.4. 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 add_tenant_access action and the tenant_id in the request body.

Normal response codes: 200

3.13.4.1. Request

This table shows the URI parameters for the add flavor access to tenant request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{flavor_id}	UUID	The flavor ID.

Example 3.85. Add flavor access to tenant: JSON request

```
{
  "addTenantAccess": {
    "tenant": "fake_tenant"
  }
}
```

3.13.4.2. Response

Example 3.86. Add flavor access to tenant: JSON response

```
{
  "flavor_access": [
    {
      "flavor_id": "10",
      "tenant_id": "fake_tenant"
    }
  ]
}
```

3.13.5. 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 `remove_tenant_access` action and the `tenant_id` in the request body.

Normal response codes: 200

3.13.5.1. Request

This table shows the URI parameters for the remove flavor access from tenant request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{flavor_id}	UUID	The flavor ID.

Example 3.87. Remove flavor access from tenant: JSON request

```
{
    "removeTenantAccess": {
        "tenant": "fake_tenant"
    }
}
```

3.13.5.2. Response

Example 3.88. Remove flavor access from tenant: JSON response

```
{
    "flavor_access": []
}
```

3.13.6. List flavors with details and access list

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/os-flavor-access/detail	Lists flavors with details. Includes access list information.

Normal response codes: 200

3.13.6.1. Request

This table shows the URI parameters for the list flavors with details and access list request:

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

This operation does not accept a request body.

3.13.6.2. Response

Example 3.89. List flavors with details and access list: 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/flavors/1",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/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/flavors/2",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/flavors/2",
          "rel": "bookmark"
        }
      ],
      "name": "m1.small",
      "ram": 1024,
      "swap": 1024,
      "vcpus": 1
    }
  ]
}
```

```
        "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/flavors/3",
    "rel": "self"
},
{
    "href": "http://openstack.example.com/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/flavors/4",
    "rel": "self"
},
{
    "href": "http://openstack.example.com/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/flavors/5",
    "rel": "self"
}
```

```
        },
        {
            "href": "http://openstack.example.com/flavors/5",
            "rel": "bookmark"
        }
    ],
    "name": "m1.xlarge",
    "ram": 16384,
    "swap": "",
    "vcpus": 8
}
]
```

3.14. 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/os-flavor-extra-specs/{flavor_id}	Creates and updates flavor extra specs.
GET	/v2.1/{tenant_id}/flavors/os-flavor-extra-specs/{flavor_id}	Lists extra specs for a flavor.
GET	/v2.1/{tenant_id}/flavors/os-flavor-extra-specs/{flavor_id}/{flavor_extra_spec_key}	Shows an extra spec for a flavor, by key.
PUT	/v2.1/{tenant_id}/flavors/os-flavor-extra-specs/{flavor_id}/{flavor_extra_spec_key}	Updates an extra spec value, by key, for a flavor.

3.14.1. Create or update flavor extra specs

Method	URI	Description
POST	/v2.1/{tenant_id}/flavors/os-flavor-extra-specs/{flavor_id}	Creates and updates flavor extra specs.

Normal response codes: 201

3.14.1.1. Request

This table shows the URI parameters for the create or update flavor extra specs request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{flavor_id}	UUID	The flavor ID.

Example 3.90. Create or update flavor extra specs: JSON request

```
{
    "extra_specs": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

3.14.1.2. Response

Example 3.91. Create or update flavor extra specs: JSON response

```
{
    "extra_specs": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

3.14.2. Show flavor extra specs

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/os-flavor-extra-specs/{flavor_id}	Lists extra specs for a flavor.

Normal response codes: 200

3.14.2.1. Request

This table shows the URI parameters for the show flavor extra specs request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{flavor_id}	UUID	The flavor ID.

Example 3.92. Show flavor extra specs: JSON request

```
{
    "extra_specs": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

3.14.2.2. Response

Example 3.93. Show flavor extra specs: JSON response

```
{
    "extra_specs": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

3.14.3. Show flavor extra specs

Method	URI	Description
GET	/v2.1/{tenant_id}/flavors/os-flavor-extra-specs/{flavor_id}/{flavor_extra_spec_key}	Shows an extra spec for a flavor, by key.

Normal response codes: 200

3.14.3.1. Request

This table shows the URI parameters for the show flavor extra specs request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{flavor_id}	UUID	The flavor ID.
{flavor_extra_spec_key}	String	The extra spec key for the flavor.

Example 3.94. Show flavor extra specs: JSON request

```
{
    "extra_specs": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

3.14.3.2. Response

Example 3.95. Show flavor extra specs: JSON response

```
{
    "key1": "value1"
}
```

3.14.4. Update flavor extra spec

Method	URI	Description
PUT	/v2.1/{tenant_id}/flavors/os-flavor-extra-specs/{flavor_id}/{flavor_extra_spec_key}	Updates an extra spec value, by key, for a flavor.

Normal response codes: 200

3.14.4.1. Request

This table shows the URI parameters for the update flavor extra spec request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{flavor_id}	UUID	The flavor ID.
{flavor_extra_spec_key}	String	The extra spec key for the flavor.

Example 3.96. Update flavor extra spec: JSON request

```
{
    "key1": "new_value1"
}
```

3.14.4.2. Response

Example 3.97. Update flavor extra spec: JSON response

```
{
    "key1": "new_value1"
}
```

3.15. Flavors manage (flavors, os-flavor-manage)

Creates and deletes flavors.

Method	URI	Description
POST	/v2.1/{tenant_id}/flavors/os-flavor-manage	Creates a flavor.
DELETE	/v2.1/{tenant_id}/flavors/os-flavor-manage/{flavor_id}	Deletes a flavor.

3.15.1. Create flavor

Method	URI	Description
POST	/v2.1/{tenant_id}/flavors/os-flavor-manage	Creates a flavor.

Normal response codes: 201

3.15.1.1. Request

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

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

Example 3.98. Create flavor: JSON request

```
{
  "flavor": {
    "name": "test_flavor",
    "ram": 1024,
    "vcpus": 2,
    "disk": 10,
    "id": "10"
  }
}
```

3.15.1.2. Response

Example 3.99. 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
  }
}
```

3.15.2. Delete flavor

Method	URI	Description
DELETE	/v2.1/{tenant_id}/flavors/os-flavor-manage/{flavor_id}	Deletes a flavor.

Normal response codes: 204

3.15.2.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{flavor_id}	UUID	The flavor ID.

Example 3.100. Delete flavor: JSON request

```
{
  "flavor": {
    "name": "test_flavor",
    "ram": 1024,
    "vcpus": 2,
    "disk": 10,
    "id": "10"
  }
}
```

3.16. Keypairs (keypairs)

Generates, imports, and deletes SSH keys.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-keypairs{?user_id}	Lists keypairs that are associated with the account.
POST	/v2.1/{tenant_id}/os-keypairs{?user_id}	Generates or imports a keypair.
DELETE	/v2.1/{tenant_id}/os-keypairs/{keypair_name}{?user_id}	Deletes a keypair.
GET	/v2.1/{tenant_id}/os-keypairs/{keypair_name}{?user_id}	Shows a keypair that is associated with the account.

3.16.1. List keypairs

Method	URI	Description
GET	/v2.1/{tenant_id}/os-keypairs{?user_id}	Lists keypairs that are associated with the account.

Normal response codes: 200

3.16.1.1. Request

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

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

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

Name	Type	Description
user_id	String <i>(Optional)</i>	The user ID of the user who runs the operation.

This operation does not accept a request body.

3.16.1.2. Response

Example 3.101. 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",
                "type": "ssh",
                "public_key": "ssh-rsa"
            }
        }
    ]
}
```

3.16.2. Create or import keypair

Method	URI	Description
POST	/v2.1/{tenant_id}/os-keypairs{?user_id}	Generates or imports a keypair.

Normal response codes: 201

3.16.2.1. Request

This table shows the URI parameters for the create or import keypair request:

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

This table shows the query parameters for the create or import keypair request:

Name	Type	Description
user_id	String <i>(Optional)</i>	The user ID of the user who runs the operation.

Example 3.102. Create or import keypair: JSON request

```
{
  "keypair": {
    "name": "keypair-d20a3d59-9433-4b79-8726-20b431d89c78",
    "type": "ssh",
    "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAAgQDx8nkQv/zgGgB4rMYmIf+6A4l6Rr+o/61HBQdW5aYd44bd8JttDCE/F/pNRr01RE+PiqSPO8nDPHw0010JeMH9gYgnnFlyY3/OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/0d3ac30bNH6Sw9JD9UZHcpSxsIbECHw== Generated-by-Nova",
    "user_id": "fake"
  }
}
```

3.16.2.2. Response

Example 3.103. 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",
    "type": "ssh",
    "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAAgQDx8nkQv/zgGgB4rMYmIf+6A4l6Rr+o/61HBQdW5aYd44bd8JttDCE/F/pNRr01RE+PiqSPO8nDPHw0010JeMH9gYgnnFlyY3/OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/0d3ac30bNH6Sw9JD9UZHcpSxsIbECHw== Generated-by-Nova",
    "user_id": "fake"
  }
}
```

3.16.3. Delete keypair

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-key-pairs/{keypair_name}{?user_id}	Deletes a keypair.

Normal response codes: 204

3.16.3.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{keypair_name}	String	The keypair name.

This operation does not accept a request body.

3.16.4. Show keypair information

Method	URI	Description
GET	/v2.1/{tenant_id}/os-key-pairs/{keypair_name} {?user_id}	Shows a keypair that is associated with the account.

Normal response codes: 200

3.16.4.1. Request

This table shows the URI parameters for the show keypair information request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{keypair_name}	String	The keypair name.

This operation does not accept a request body.

3.16.4.2. Response

Example 3.104. Show keypair information: 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",
    "type": "ssh",
    "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQ1HTrHCbb9NawNLSV8N6tSa8i637+EC2dA+lsdHHfQ1T54t
+N0nHhJP1KWDLhc579j87vp6RDFriFJ/smsTnDnf64012z0kBaJpJPH2zXrBkZFK6q2rmxydURzX/
z0yLSCP77SFJ0fdXWH2hMsAusflGyryHGX20n
+mZK6mDrxVzGxEz228dwQ5G7Az5OoZDWygH2pqPvKjkifRw0jwUKf3BbkP0QvANACOk26cv16mNFpFJfI1N3OC51UsZQ
qup58J5kf1Nm7I61sy1mJon6SGqNUSfoQagqtBH6vd/tU1jn1wZ03uUroAL 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
  }
}
```

3.17. Limits (limits)

Shows all global and rate limit information.

Method	URI	Description
GET	/v2.1/{tenant_id}/limits	Shows global and rate limit information.

3.17.1. Show global and rate limits

Method	URI	Description
GET	/v2.1/{tenant_id}/limits	Shows global and rate limit information.

Normal response codes: 200

3.17.1.1. Request

This table shows the URI parameters for the show global and rate limits request:

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

This operation does not accept a request body.

3.17.1.2. Response

Example 3.105. Show global and rate limits: JSON response

```
{
  "limits": {
    "absolute": {
      "maxImageMeta": 128,
      "maxPersonality": 5,
      "maxPersonalitySize": 10240,
      "maxSecurityGroupRules": 20,
      "maxSecurityGroups": 10,
      "maxServerMeta": 128,
      "maxTotalCores": 20,
      "maxTotalFloatingIps": 10,
      "maxTotalInstances": 10,
      "maxTotalKeypairs": 100,
      "maxTotalRAMSize": 51200,
      "maxServerGroups": 10,
      "maxServerGroupMembers": 10,
      "totalCoresUsed": 0,
      "totalInstancesUsed": 0,
      "totalRAMUsed": 0,
      "totalSecurityGroupsUsed": 0,
      "totalFloatingIpsUsed": 0,
      "totalServerGroupsUsed": 0
    },
    "rate": []
  }
}
```

3.18. Extensions (extensions)

Available extensions.

Method	URI	Description
GET	/v2.1/extensions	Lists available extensions.

Method	URI	Description
GET	/v2.1/extensions/{alias}	Shows details for an extension.

3.18.1. List extensions

Method	URI	Description
GET	/v2.1/extensions	Lists available extensions.

Normal response codes: 200203

3.18.1.1. Request

This operation does not accept a request body.

3.18.1.2. Response

Example 3.106. 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-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\ninclude: 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.\n2. Add availability zones describing.\n      ",
        "links": [],
        "name": "AvailabilityZone",
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "updated": "2014-12-03T00:00:00Z"
    }
```

```
        "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 cloupipe instances.\n\nWhen 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 cloupipe.\nAt 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"
    }
]
```

```
        "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"
}
]
}

```

Example 3.107. 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"
        }
    ]
}
```

```
        "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-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\nActions 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"
    }
```

```
        "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.\n2. 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\nWhen running with the Vlan network mode, you need a mechanism to route\nfrom the public Internet to your vlans. This mechanism is known as a\ncloudpipe.\n\nAt the time of creating this class, only OpenVPN is supported. Support for\na SSH Bastion host is forthcoming.",  
    "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"  
}
```

```
        "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"
    }
```

```
        "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"
    }
]
```

3.18.2. Get extension

Method	URI	Description
GET	/v2.1/extensions/{alias}	Shows details for an extension.

Extensions introduce features and vendor-specific functionality to the API without requiring a version change.

Normal response codes: 200203

3.18.2.1. Request

This table shows the URI parameters for the get extension request:

Name	Type	Description
{alias}	String	An alias, which is a name for a pointer to a resource. For example, a named extension.

This operation does not accept a request body.

3.18.2.2. Response

Example 3.108. Get extension: JSON response

```
{
  "extension": {
    "updated": "2011-09-27T00:00:00+00:00",
    "name": "DiskConfig",
    "links": [],
    "namespace": "http://docs.openstack.org/compute/ext/disk_config/api/v1.1",
    "alias": "OS-DCF",
    "description": "Disk Management Extension."
  }
}
```

Example 3.109. Get extension: JSON response

```
{
  "extension": {
    "updated": "2011-09-27T00:00:00+00:00",
    "name": "DiskConfig",
    "links": [],
    "namespace": "http://docs.openstack.org/compute/ext/disk_config/api/v1.1",
    "alias": "OS-DCF",
    "description": "Disk Management Extension."
  }
}
```

3.19. Images

Lists, gets details for, and deletes images. Also sets, lists, gets 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.

If you set the image size policy in the `policy.json` file, the `OS-EXT-IMG-SIZE:size` extended attribute is visible.

Method	URI	Description
GET	<code>/v2.1/images{ ?changes-since,server,name,status,type,limit,marker}</code>	Lists IDs, names, and links for available images.
GET	<code>/v2.1/images/detail{ ?changes-since,server,name,status,type,limit,marker}</code>	Lists all details for available images.
GET	<code>/v2.1/images/{image_id}</code>	Gets details for an image.
DELETE	<code>/v2.1/images/{image_id}</code>	Deletes an image.

3.19.1. List images

Method	URI	Description
GET	/v2.1/images{ ?changes-since,server,name,status,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)

3.19.1.1. Request

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

Name	Type	Description
changes-since	DateTime <i>(Optional)</i>	<p>The date and time when the image last changed status.</p> <p>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 <code>changes-since</code> time. If data has not changed since the <code>changes-since</code> time, the call returns an empty list.</p> <p>To enable you to keep track of changes, this filter also displays images that were deleted if the <code>changes-since</code> value specifies a date in the last 30 days. Items deleted more than 30 days ago might be returned, but it is not guaranteed.</p> <p>The date and time stamp format is ISO 8601:</p> <p style="background-color: #f0f0f0; padding: 2px;"><code>CCYY-MM-DDThh:mm:ss±hh:mm</code></p> <p>The <code>±hh:mm</code> value, if included, returns the time zone as an offset from UTC.</p> <p>For example, <code>2015-08-27T09:49:58-05:00</code>.</p> <p>If you omit the time zone, the UTC time zone is assumed.</p>
server	AnyURI <i>(Optional)</i>	Name of the server in URL format.
name	String <i>(Optional)</i>	Name of the image as a string.
status	ImageStatus <i>(Optional)</i>	Value of the image statuses. For example, you can filter on ACTIVE.
type	String <i>(Optional)</i>	Value of the type of image, such as snapshot or backup. Possible values: snapshot, backup. Default: ALL.
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

3.19.1.2. Response

Example 3.110. 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"
}
]
}

```

Example 3.111. 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.

3.19.2. List images details

Method	URI	Description
GET	/v2.1/images/detail{?changes-since,server,name,status,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)

3.19.2.1. Request

This table shows the query parameters for the list images details request:

Name	Type	Description
changes-since	DateTime <i>(Optional)</i>	<p>The date and time when the image last changed status.</p> <p>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 <code>changes-since</code> time. If data has not changed since the <code>changes-since</code> time, the call returns an empty list.</p> <p>To enable you to keep track of changes, this filter also displays images that were deleted if the <code>changes-since</code> value specifies a date in the last 30 days. Items deleted more than 30 days ago might be returned, but it is not guaranteed.</p> <p>The date and time stamp format is ISO 8601:</p> <p style="background-color: #f0f0f0; padding: 2px;"><code>CCYY-MM-DDThh:mm:ss±hh:mm</code></p> <p>The <code>±hh:mm</code> value, if included, returns the time zone as an offset from UTC.</p> <p>For example, <code>2015-08-27T09:49:58-05:00</code>.</p> <p>If you omit the time zone, the UTC time zone is assumed.</p>
server	AnyURI <i>(Optional)</i>	Name of the server in URL format.
name	String <i>(Optional)</i>	Name of the image as a string.
status	ImageStatus <i>(Optional)</i>	Value of the image statuses. For example, you can filter on ACTIVE.
type	String <i>(Optional)</i>	Value of the type of image, such as snapshot or backup.
limit	Int <i>(Optional)</i>	Possible values: snapshot, backup. Default: ALL.
		Requests a page size of returned items from the query. 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.

Name	Type	Description
marker	String <i>(Optional)</i>	Specifies 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.

3.19.2.2. Response

Example 3.112. 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/v2.1/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
          "rel": "bookmark"
        }
      ]
    }
  ]
}
```

```
{  
    "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-bed1-0b816eaec379",  
    "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"
    }
],
{
    "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"
}
```

```
        "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"
    }
]
}
```

Example 3.113. 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-bed1-0b816eaec379",
    "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"
    }
],
"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.

3.19.3. Get image details

Method	URI	Description
GET	/v2.1/images/{image_id}	Gets 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)

3.19.3.1. Request

This table shows the URI parameters for the get image details request:

Name	Type	Description
{image_id}	UUID	The UUID for the image.

This operation does not accept a request body.

3.19.3.2. Response

Example 3.114. Get 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"
    }
}
```

Example 3.115. Get 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"
    }
}
```

3.19.4. Delete image

Method	URI	Description
DELETE	/v2.1/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)

3.19.4.1. Request

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

Name	Type	Description
{image_id}	UUID	The UUID for the image.

This operation does not accept a request body.

3.20. Image metadata

Shows details for, sets, updates, and deletes image metadata or metadata items.

Method	URI	Description
GET	/v2.1/images/{image_id}/metadata	Shows metadata for an image.
PUT	/v2.1/images/{image_id}/metadata	Creates or replaces metadata for an image.
POST	/v2.1/images/{image_id}/metadata	Updates metadata items, by key, for an image.
GET	/v2.1/images/{image_id}/metadata/{key}	Shows details for a metadata item, by key, for an image.
PUT	/v2.1/images/{image_id}/metadata/{key}	Creates or updates a metadata item, by key, for an image.
DELETE	/v2.1/images/{image_id}/metadata/{key}	Deletes a metadata item, by key, for an image.

3.20.1. Show image metadata

Method	URI	Description
GET	/v2.1/images/{image_id}/metadata	Shows metadata for an image.

Normal response codes: 200 203

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

3.20.1.1. Request

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

Name	Type	Description
{image_id}	UUID	The UUID for the image.

This operation does not accept a request body.

3.20.1.2. Response

Example 3.116. Show image metadata: JSON response

```
{
  "metadata": {
    "architecture": "x86_64",
    "auto_disk_config": "True",
    "kernel_id": "nokernel",
    "ramdisk_id": "nokernel"
  }
}
```

Example 3.117. 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.

3.20.2. Create or replace image metadata

Method	URI	Description
PUT	/v2.1/images/{image_id}/metadata	Creates or replaces metadata for an image.

Replaces items that match specified keys. If you omit a key that already exists, this key retains its value.

If this operation exceeds the metadata items quota, the API throws an `overLimit` (413) fault.

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)

3.20.2.1. Request

This table shows the URI parameters for the create or replace image metadata request:

Name	Type	Description
{image_id}	UUID	The UUID for the image.

Example 3.118. Create or replace image metadata: JSON request

```
{
  "metadata": {
    "auto_disk_config": "True",
    "Label": "Changed"
  }
}
```

3.20.2.2. Response

Example 3.119. Create or replace image metadata: JSON response

```
{
  "metadata": {
    "Label": "Changed",
    "auto_disk_config": "True"
  }
}
```

3.20.3. Update image metadata items

Method	URI	Description
POST	/v2.1/images/{image_id}/metadata	Updates metadata items, by key, for an image.

Replaces items that match the keys and does not modify items not in the request.

If this operation exceeds the metadata items quota, the API throws an `overLimit` (413) fault.

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)

3.20.3.1. Request

This table shows the URI parameters for the update image metadata items request:

Name	Type	Description
{image_id}	UUID	The UUID for the image.

Example 3.120. Update image metadata items: JSON request

```
{
  "metadata": {
    "kernel_id": "False",
    "Label": "UpdatedImage"
  }
}
```

3.20.3.2. Response

Example 3.121. 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.

3.20.4. Show image metadata item details

Method	URI	Description
GET	/v2.1/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)

3.20.4.1. Request

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

Name	Type	Description
{image_id}	UUID	The UUID for the image.
{key}	Image Metadata Key	The metadata item key, as a string. Maximum length is 255 characters.

This operation does not accept a request body.

3.20.4.2. Response

Example 3.122. Show image metadata item details: JSON response

```
{
  "metadata": {
    "architecture": "x86_64",
    "auto_disk_config": "True",
    "kernel_id": "nokernel",
    "ramdisk_id": "nokernel"
  }
}
```

Example 3.123. Show image metadata item details: JSON response

```
{
  "metadata": {
    "architecture": "x86_64",
    "auto_disk_config": "True",
    "kernel_id": "nokernel",
    "ramdisk_id": "nokernel"
  }
}
```

3.20.5. Create or update image metadata item

Method	URI	Description
PUT	/v2.1/images/{image_id}/metadata/{key}	Creates or updates a metadata item, by key, for an image.

If this operation exceeds the metadata items quota, the API throws an `overLimit` (413) fault.

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)

3.20.5.1. Request

This table shows the URI parameters for the create or update image metadata item request:

Name	Type	Description
{image_id}	UUID	The UUID for the image.
{key}	Image Metadata Key	The metadata item key, as a string. Maximum length is 255 characters.

Example 3.124. Create or update image metadata item: JSON request

```
{
  "metadata": {
    "auto_disk_config": "True",
    "Label": "Changed"
  }
}
```

3.20.5.2. Response

Example 3.125. Create or update image metadata item: JSON response

```
{
  "metadata": {
    "Label": "Changed",
    "auto_disk_config": "True"
  }
}
```

3.20.6. Delete image metadata item

Method	URI	Description
DELETE	/v2.1/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)

3.20.6.1. Request

This table shows the URI parameters for the delete image metadata item request:

Name	Type	Description
{image_id}	UUID	The UUID for the image.
{key}	Image Metadata Key	The metadata item key, as a string. Maximum length is 255 characters.

This operation does not accept a request body.

3.21. 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/{id}	Updates an agent build.

3.21.1. Create agent build

Method	URI	Description
POST	/v2.1/{tenant_id}/os-agents	Creates an agent build.

Normal response codes: 201

3.21.1.1. Request

This table shows the URI parameters for the create agent build request:

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

Example 3.126. Create agent build: JSON request

```
{
  "agent": {
    "hypervisor": "hypervisor",
    "os": "os",
    "architecture": "x86",
    "version": "8.0",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "url": "http://example.com/path/to/resource"
  }
}
```

3.21.1.2. Response

Example 3.127. 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"
  }
}
```

3.21.2. List agent builds

Method	URI	Description
GET	/v2.1/{tenant_id}/os-agents	Lists agent builds.

Normal response codes: 200

3.21.2.1. Request

This table shows the URI parameters for the list agent builds request:

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

This operation does not accept a request body.

3.21.2.2. Response

Example 3.128. 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"
    }
  ]
}
```

3.21.3. Delete agent build

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-agents	Deletes an existing agent build.

Normal response codes: 202

3.21.3.1. Request

This table shows the URI parameters for the delete agent build request:

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

This operation does not accept a request body.

3.21.4. Update agent build

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-agents/{id}	Updates an agent build.

Normal response codes: 200

3.21.4.1. Request

This table shows the URI parameters for the update agent build request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{id}	UUID	The agent build ID.

Example 3.129. Update agent build: JSON request

```
{
  "para": {
    "url": "http://example.com/path/to/resource",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "version": "7.0"
  }
}
```

3.21.4.2. Response

Example 3.130. Update agent build: JSON response

```
{
  "agent": {
    "agent_id": "1",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "url": "http://example.com/path/to/resource",
    "version": "7.0"
  }
}
```

3.22. 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 aggregates id, name, and availability_zone for an aggregate.
GET	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Shows the details of an aggregate, hosts and metadata included.
PUT	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Updates either or both the name and availability zone for an aggregate.

Method	URI	Description
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.

3.22.1. Create aggregate

Method	URI	Description
POST	/v2.1/{tenant_id}/os-aggregates	Creates an aggregate in an availability zone.

Normal response codes: 200

3.22.1.1. Request

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

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

Example 3.131. Create aggregate: JSON request

```
{
    "aggregate": {
        "name": "name",
        "availability_zone": "nova"
    }
}
```

3.22.1.2. Response

Example 3.132. 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
    }
}
```

3.22.2. List aggregates

Method	URI	Description
GET	/v2.1/{tenant_id}/os-aggregates	Lists aggregates id, name, and availability_zone for an aggregate.

Normal response codes: 200

3.22.2.1. Request

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

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

Example 3.133. List aggregates: JSON request

```
{
  "aggregate": {
    "name": "name",
    "availability_zone": "nova"
  }
}
```

3.22.2.2. Response

Example 3.134. 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
    }
  ]
}
```

3.22.3. Show aggregate details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Shows the details of an aggregate, hosts and metadata included.

Normal response codes: 200

3.22.3.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{aggregate_id}	Int	The aggregate ID.

Example 3.135. Show aggregate details: JSON request

```
{
    "aggregate": {
        "name": "name",
        "availability_zone": "nova"
    }
}
```

3.22.3.2. Response

Example 3.136. 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
    }
}
```

3.22.4. 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

3.22.4.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{aggregate_id}	Int	The aggregate ID.

Example 3.137. Update aggregate: JSON request

```
{
    "aggregate": {
        "name": "newname",
        "availability_zone": "nova2"
    }
}
```

3.22.4.2. Response

Example 3.138. 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"
    }
}
```

3.22.5. 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

3.22.5.1. Request

This table shows the URI parameters for the add host request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{aggregate_id}	Int	The aggregate ID.

Example 3.139. Add host: JSON request

```
{
    "add_host": {
        "host": "21549b2f665945baaa7101926a00143c"
    }
}
```

3.22.5.2. Response

Example 3.140. 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
    }
}
```

3.22.6. 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 add_metadata action in the request body.

Normal response codes: 200

3.22.6.1. Request

This table shows the URI parameters for the create or update aggregate metadata request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{aggregate_id}	Int	The aggregate ID.

Example 3.141. Create or update aggregate metadata: JSON request

```
{
    "set_metadata": {
        "metadata": {
            "key": "value"
        }
    }
}
```

3.22.6.2. Response

Example 3.142. 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
    }
}
```

3.22.7. 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

3.22.7.1. Request

This table shows the URI parameters for the remove host request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{aggregate_id}	Int	The aggregate ID.

Example 3.143. Remove host: JSON request

```
{
    "remove_host": {
        "host": "bf1454b3d71145d49fca2101c56c728d"
    }
}
```

3.22.7.2. Response

Example 3.144. 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
    }
}
```

3.23. 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.

3.23.1. 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

3.23.1.1. Request

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

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

Example 3.145. 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"
    }
  }
}
```

3.23.1.2. Response

Example 3.146. Create assisted volume snapshots: JSON response

```
{
  "snapshot": {
    "id": 100,
    "volumeId": "521752a6-acf6-4b2d-bc7a-119f9148cd8c"
  }
}
```

3.23.2. 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

3.23.2.1. Request

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

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

This table shows the query parameters for the delete assisted volume snapshot request:

Name	Type	Description
delete_info	String <i>(Required)</i>	Information for snapshot deletion. Include the ID of the associated volume. For example: DELETE /os-assisted-volume-snapshots?delete_info='{"volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c"}'

This operation does not accept a request body.

3.24. Attach Interfaces (os-attach-interfaces)

Creates, lists, gets details for, and deletes port interfaces.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces	Creates a port interface and uses it to attach a port to a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces	Lists port interfaces that are attached to a server.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces/{attachment_id}	Shows details for a port interface that is attached to a server.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces/{attachment_id}	Detaches a port interface.

3.24.1. Create interface

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces	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, ...), computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), badMediaType (415), NetworkNotFound (400)

3.24.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.

Example 3.147. Create interface: JSON request

```
{
  "interfaceAttachment": {
    "port_id": "ce531f90-199f-48c0-816c-13e38010b442"
  }
}
```

3.24.1.2. Response

Example 3.148. 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"
  }
}
```

3.24.2. List interfaces

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces	Lists port interfaces that are attached to a server.

Normal response codes: 200

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

3.24.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.

This operation does not accept a request body.

3.24.2.2. Response

Example 3.149. List 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"
        }
    ]
}
```

3.24.3. Show port interface details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces/{attachment_id}	Shows details for a port interface that is attached to a server.

Normal response codes: 200

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

3.24.3.1. Request

This table shows the URI parameters for the show port interface details request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.
{attachment_id}	UUID	The interface ID.

This operation does not accept a request body.

3.24.3.2. Response

Example 3.150. 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"
    }
}
```

3.24.4. Detach interface

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-attach-interfaces/{attachment_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)

3.24.4.1. Request

This table shows the URI parameters for the detach interface request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.
{attachment_id}	UUID	The interface ID.

This operation does not accept a request body.

3.25. Availability zones (os-availability-zone)

Gets 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.

3.25.1. Get availability zone information

Method	URI	Description
GET	/v2.1/{tenant_id}/os-availability-zone	Gets availability zone information.

Normal response codes: 200

3.25.1.1. Request

This table shows the URI parameters for the get availability zone information request:

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

This operation does not accept a request body.

3.25.1.2. Response

Example 3.151. Get availability zone information: JSON response

```
{
    "availabilityZoneInfo": [
        {
            "zoneState": {
                "available": true
            },
            "hosts": null,
            "zoneName": "nova"
        }
    ]
}
```

3.25.2. 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

3.25.2.1. Request

This table shows the URI parameters for the get detailed availability zone information request:

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

This operation does not accept a request body.

3.25.2.2. Response

Example 3.152. 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"
}
]
}

```

3.26. 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.

3.26.1. 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)

3.26.1.1. Request

This table shows the URI parameters for the add bare metal node request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.

Example 3.153. 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 3.154. 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",
    "ram_gb": 8
  }
}
```

```
        "terminal_port": 8000
    }
}
```

3.26.1.2. Response

Example 3.155. 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 3.156. 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"
  }
}
```

3.26.2. 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)

3.26.2.1. Request

This table shows the URI parameters for the list bare metal nodes request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.

This operation does not accept a request body.

3.26.2.2. Response

Example 3.157. 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"
    }
  ]
}
```

3.26.3. 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)

3.26.3.1. Request

This table shows the URI parameters for the add interface to bare metal node request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.

Example 3.158. Add interface to bare metal node: JSON request

```
{
  "add_interface": {
    "address": "aa:aa:aa:aa:aa:aa"
  }
}
```

3.26.3.2. Response

Example 3.159. Add interface to bare metal node: JSON response

```
{
  "interface": {
    "address": "aa:aa:aa:aa:aa:aa",
    "datapath_id": null,
    "id": 1,
    "port_no": null
  }
}
```

3.26.4. 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)

3.26.4.1. Request

This table shows the URI parameters for the delete interface from bare metal node request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.

Example 3.160. Delete interface from bare metal node: JSON request

```
{
    "remove_interface": {
        "address": "aa:aa:aa:aa:aa:aa"
    }
}
```

3.26.5. 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)

3.26.5.1. Request

This table shows the URI parameters for the show bare metal node details request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.
{node_id}	String	Node ID.

This operation does not accept a request body.

3.26.5.2. Response

Example 3.161. 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,
    "volumes": [
      {
        "id": 1,
        "size_gb": 100
      }
    ]
  }
}
```

```
        "uuid": "c862b836-c7c1-4f7f-8081-6766fa9cf38b"  
    }  
}
```

3.26.6. 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.

3.26.6.1. Request

This table shows the URI parameters for the delete bare metal node request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.
{node_id}	String	Node ID.

This operation does not accept a request body.

3.27. Servers with block device mapping format (servers, os-block-device-mapping)

Creates a server with a block device mapping.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers	Creates a server with a block device mapping.

3.27.1. Create server with block device mapping

Method	URI	Description
POST	/v2.1/{tenant_id}/servers	Creates a server with a block device mapping.

To define the block device mapping, you can include either a `block_device_mapping` or `block_device_mapping_v2` object in the request body. The `block_device_mapping_v2` object is preferred.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), UnprocessableEntity (422), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), serverCapacityUnavailable (503)

3.27.1.1. Request

This table shows the URI parameters for the create server with block device mapping request:

Name	Type	Description
{tenant_id}	String	The tenant ID in a multi-tenancy cloud.

Example 3.162. Create server with block device mapping: JSON request

```
{
    "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"
        },
        "personality": [
            {
                "path": "/etc/banner.txt",
                "contents": "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcyBpbBqdXN0IHN1Y2ggYSBkaXJ1Y3Rpb2"
            }
        ],
        "block_device_mapping_v2": [
            {
                "device_name": "/dev/sdb1",
                "source_type": "blank",
                "destination_type": "local",
                "delete_on_termination": "True",
                "guest_format": "swap",
                "boot_index": "-1"
            },
            {
                "device_name": "/dev/sda1",
                "source_type": "volume",
                "volume_id": "volid-12345678901234567890123456789012"
            }
        ]
    }
}
```

```

        "destination_type": "volume",
        "uuid": "fake-volume-id-1",
        "boot_index": "0"
    }
]
}
}
```

3.27.1.2. Response

Example 3.163. Create server with block device mapping: JSON response

```
{
  "server": {
    "adminPass": "N4x7wFX6iN8D",
    "id": "babd1af0-4fc6-4529-b32f-aad69811ccf5",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/babd1af0-4fc6-4529-b32f-aad69811ccf5",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/babd1af0-4fc6-4529-b32f-aad69811ccf5",
        "rel": "bookmark"
      }
    ]
  }
}
```

3.28. Cells (os-cells, capacities)

Adds neighbor cells, lists neighbor cells, and gets 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.

3.28.1. 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)

3.28.1.1. Request

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

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

This operation does not accept a request body.

3.28.1.2. Response

Example 3.164. 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": "cell14",
            "rpc_host": null,
            "rpc_port": null,
            "type": "parent",
            "username": "username4"
        }
    ]
}
```

3.28.2. 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)

3.28.2.1. Request

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

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

This operation does not accept a request body.

3.28.2.2. Response

Example 3.165. List cells with details: JSON response

```
{  
    "cells": []  
}
```

3.28.3. 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)

3.28.3.1. Request

This table shows the URI parameters for the show cell data request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{cell_id}	UUID	The cell ID.

This operation does not accept a request body.

3.28.3.2. Response

Example 3.166. Show cell data: JSON response

```
{
    "cell": {
        "name": "cell3",
        "rpc_host": null,
        "rpc_port": null,
        "type": "child",
        "username": "username3"
    }
}
```

3.28.4. 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)

3.28.4.1. Request

This table shows the URI parameters for the show cell capacities request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{cell_id}	UUID	The cell ID.

This operation does not accept a request body.

3.28.4.2. Response

Example 3.167. 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
        }
      }
    }
  }
}
```

3.29. 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.

3.29.1. Create certificate

Method	URI	Description
POST	/v2.1/{tenant_id}/os-certificates	Creates a certificate.

Normal response codes: 201

3.29.1.1. Request

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

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

This operation does not accept a request body.

3.29.1.2. Response

Example 3.168. 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-----\nMIICNDCCAZ0CAgP6MA0GCSqGSIb3DQEBAUAME4xEjAQBgNVBAoTCU5PVkEgUk9P\\nVDEWMBQGA1UEBxMNTW91bnRhaW4gVmlldzETMBEGA1UECBMKQ2FsaWZvcm5pYTEL\\nMAkGA1UEBhMCVVMwHhcNMTMwODEyMDcyMDMwWhcNMTQwODEyMDcyMDMwWjB2MQsw\\nCQYDVQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcm5pYTESMBAGA1UECgwJT3B1b1N0\\nYWNrMRAwDgYDVQQLDAdOb3ZhRGV2MSwwKgYDVQQDDCNvcGVuc3RhY2stZmFrZS0y\\nMDEzLTA4LTEyVDA3OjIwOjMwWjCBnzANBggkqhkiG9w0BAQEFAAOBJQAwgYkCgYE\\nrP+x0e1UTjVsNLSPCwRQJaPiTwJMTyZZvfP969oYwjaqY0JyH4hPOuznn45EKtO4\\"
    }
}
```

```
n1HsgQfhIAleRTBZi8SHU8kC1h1DZYfC+/9iNn0uqagc4on+HIfzmbh0K1RqQDmDC\
nJ0mO6Ggb6fPGsHzaxSBmm4Xq9cmn3u4WsVGgTeOVmN8CAwEAATANBgkqhkiG9w0B\
nAQQFAAOBgQAVQspxzDKv2.1M9Fkd+KuDDEf3iApyXC2YE+s90izDv410ePBPaTBJ6F
\n1BBA/1oHRyS1rpOtjeHmVEqNSilTxI0Eawv2rzh4AsUFGYmCLbr9ETweGMkMPQOT\
nbrxmcDTuA3iKHT1k6CAvkIGOSR0HNxVmQstYOa1WzulHxngLDnUpyg==\n-----END
CERTIFICATE----\n",
    "private_key": "-----BEGIN RSA PRIVATE KEY-----\
nMIICXgIBAAKBgQC.../7HR7VRONWw0tI8LBFAlo+JPAkxPJlm98/3r2hjCNqpjQnIf\
niE867OefjkQq07iUeyBB+EgCV5FMFmLxIdTyQLWGUNlh8L7/2I2fS6pqBziif4ch\
n/OZuHQqVGpAOYMIk6Y7oaBvp88awfNrFIGabher1yafe7haxUaBN45WY3wIDAQAB\
nAoGBAIrcr2I/KyWf0hw4Nn10V9TuyE/9Gz2JHg3QFKjfJox2DqygADT5WAeHc6Bq\
nNKnf0NA2SL1LSp...+ql01tv0w4VjE5TF6OH...IzHuTTnXggG6vuA8rxp6L24HtkAcc\
n0CBno9ggSX6jVornJPBfxpkwITYSvH57BUFVD7ovbPyWGzS5AkEA1JeUtL6zxwps\
nWRr1aJ8I112uQk/RUIvSZOU61s+B190zvHikFy8LD8CI6vvBmjC/IzuZVedufjq...\
n4vX82uDO3QJBANBS...h2b2dyB4AGVFY9vXM...rtALAspJHbLHy+zTKx1GPFiuz7Se3ps\
n8Kehz4C/CBXgQkk194dwFSGE19/PQfyJROsCQQCFFDJZhrtBUMwMZ2zSRiN5BUGt\
nbwuncS+OS1Su3Yz5VRYq2BZYEPHKtYrAFkLWQ8eRwTaWaN5pFE/fb38OgQXdAkA4\
nDm0W/K0z1HbuyUxEpNQ28/6mBi0ktiWvLT0tioq6sYmXLwZA/D2JrhXrG/xt/o13\
nr8jqr...fNRsLByLhAgh0N/AkEA12eR0O971TEgFNqzIQwVmIA...n9mB03cnf3tycv1DU\
nm6eb2CS242y4QalfCCAEjxoJURdfsm3/D1iFo00X+iWF+A==\n-----END RSA PRIVATE
KEY----\n"
}
}
```

3.29.2. Show certificate details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-certificates/{certificate_id}	Shows details for a certificate.

Normal response codes: 200

3.29.2.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{certificate_id}	UUID	The certificate ID.

This operation does not accept a request body.

3.29.2.2. Response

Example 3.169. Show certificate details: JSON response

```
{
  "certificate": {
    "data": "-----BEGIN CERTIFICATE-----\\nMIICyzCCAjSgAwIBAgIJAJ8zSIXUp/m4MA0GCSqGSIb3DQEBAUAME4xEjAQBgNV\\nBAoTCU5PVkEgUk9PVDEWMBQGA1UEBxMNTW91bnRhaW4qVmlldzETMBEGA1UECBMK\\nQ2FsaWZvcm5pYTELMAkGA1UEBhMCVVMwHhcNMTIxMDE3MDEzMzM5WhcNMTMzMDE3\\nMDEzMzM5WjBOMRIwEAYDVQQKEw1OT1ZBIFJPT1QxFjAUBgNVBAcTDU1vdW50YWlu\\nIFZpZXcxEzARBgNVBAgTCkNhbg1mb3JuaWEExCzAJBgNVBAYTA1VTMIGfMA0GCSqG\\nStb3DQEBAQUAA4GNADCBiQKbgQDXW4QfQQxJG4MqurqK8nU/Lge0mfNKxXj/Gwvg\\n2sQVwxzmKfoxih8Nm6yt0yHMNjhoji1uoWI03TXUnPZRASmsypGKZeBd7Y1ZOCPB\\nXGZVGrQm+PB2kZU+3cD8fVKcueMLLeZ+Lrt5d0njnoKh5xjqMlfFPimHMba4OL6\\nTnYzPQIDAQABo4GwMIGtMAwGA1UdEwQFMAMBAf8wHQYDVR0OBBYEFKyoKu4SMOFM\\ngx5Ec7p0nrCkabvxMH4GA1UdIwR3MHWAFKyoKu4SMOFMgx5Ec7p0nrCkabvxoVKx\\nUDBOMRIwEAYDVQQKEw1OT1ZBIFJPT1QxFjAUBgNVBAcTDU1vdW50YWluIFZpZXcx\\nEZARBgNVBAgTCkNhbg1mb3JuaWEExCzAJBgNVBAYTA1VTggkAnzNIjFSn+bgwDQYJ\\nKoZIhvvcNAQEEBQAdgYEAXuvXlulo/SVvykSLhhW8QiAY00yzN/eDzYmZGomgiuoO\\n\\x+ayVzbrz1UWZnBD+lC4h112iELSmf22LjLoF+s/9NyPqHxGL3FrfatBkndaiF8\\nAx/\\nTMEyCP17IQWi+3zzatqOKHSHiG7a9SGn/7o2aNTIWKVulfy5GvmbBjBM/0UE=\\n-----END\\nCERTIFICATE-----\\n",
    "private_key": null
  }
}
```

3.30. 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 cloupipe instance.

3.30.1. 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)

3.30.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

This operation does not accept a request body.

3.30.1.2. Response

Example 3.170. 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"
    }
  ]
}
```

3.30.2. Create cloupipe

Method	URI	Description
POST	/v2.1/{tenant_id}/os-cloupipe	Creates a cloupipe.

Normal response codes: 200

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

3.30.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

Example 3.171. Create cloupipe: JSON request

```
{
  "cloupipe": {
    "project_id": "059f21e3-c20e-4efc-9e7a-eba2ab3c6f9a"
  }
}
```

This operation does not accept a request body.

3.30.2.2. Response

Example 3.172. Create cloupipe: JSON response

```
{
  "instance_id": "1e9b8425-34af-488e-b969-4d46f4a6382e"
}
```

3.30.3. Update cloupipe

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 cloupipe 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)

3.30.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

Example 3.173. Update cloupipe: JSON request

```
{
  "configure_project": {
    "vpn_ip": "192.168.1.1",
    "vpn_port": "2000"
  }
}
```

This operation does not accept a request body.

3.31. Server consoles (servers, os-consoles, os-console-auth-token)

Manages server consoles.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/ {server_id}/os-consoles	Creates a console for a server instance.
GET	/v2.1/{tenant_id}/servers/ {server_id}/os-consoles	Lists all consoles for a server instance.
GET	/v2.1/{tenant_id}/servers/ {server_id}/os-con- soles/{console_id}	Shows details for a console for a server instance.
DELETE	/v2.1/{tenant_id}/servers/ {server_id}/os-con- soles/{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.

3.31.1. Create console for server

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-consoles	Creates a console for a server instance.

Normal response codes: 200

3.31.1.1. Request

This table shows the URI parameters for the create console for server request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.31.2. Lists consoles for server

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-consoles	Lists all consoles for a server instance.

Normal response codes: 200

3.31.2.1. Request

This table shows the URI parameters for the lists consoles for server request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.31.2.2. Response

Example 3.174. List consoles: JSON response

```
{
    "consoles": [
        {
            "console": {
                "console_type": "fake",
                "id": 1
            }
        }
    ]
}
```

3.31.3. Show console details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-consoles/{console_id}	Shows details for a console for a server instance.

Normal response codes: 200

3.31.3.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.
{console_id}	UUID	The ID for the console.

This operation does not accept a request body.

3.31.3.2. Response

Example 3.175. List consoles: JSON response

```
{
  "console": {
    "console_type": "fake",
    "host": "fake",
    "id": 1,
    "instance_name": "instance-00000001",
    "password": "C4jBpJ6x",
    "port": 5999
  }
}
```

3.31.4. Delete console

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-soles/{console_id}	Deletes a console for a server instance.

Normal response codes: 202

3.31.4.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.
{console_id}	UUID	The ID for the console.

This operation does not accept a request body.

3.31.5. 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

3.31.5.1. Request

This table shows the URI parameters for the show console authentication token request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.31.5.2. Response

Example 3.176. List consoles: JSON response

```
{
  "console": {
    "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
    "host": "localhost",
    "port": 5900,
    "internal_access_path": "51af38c3-555e-4884-a314-6c8cdde37444"
  }
}
```

3.32. 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.

3.32.1. 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)

3.32.1.1. Request

This table shows the URI parameters for the show fixed ip details request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{fixed_ip}	String	The fixed IP of interest to you.

This operation does not accept a request body.

3.32.1.2. Response

Example 3.177. Show fixed IP details: JSON response

```
{
    "fixed_ip": {
        "address": "192.168.1.1",
        "cidr": "192.168.1.0/24",
        "host": "host",
        "hostname": "openstack"
    }
}
```

3.32.2. 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)

3.32.2.1. Request

This table shows the URI parameters for the reserve or release a fixed ip request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{fixed_ip}	String	The fixed IP of interest to you.

Example 3.178. Reserve or release a fixed IP: JSON request

```
{
    "reserve": null
}
```

3.33. Floating IP DNS records (os-floating-ip-dns)

Manages DNS records associated with IP addresses allocated by the floating IPs extension. Requests are dispatched to a DNS driver 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.

3.33.1. 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

3.33.1.1. Request

This table shows the URI parameters for the list dns domains request:

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

This operation does not accept a request body.

3.33.1.2. Response

Example 3.179. List DNS domains: JSON response

```
{
    "domain_entries": [
        {
            "availability_zone": null,
            "domain": "domain1.example.org",
            "project": "project1",
            "scope": "public"
        }
    ]
}
```

3.33.2. 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

3.33.2.1. Request

This table shows the URI parameters for the create or update dns domain request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{domain}	String	Registered DNS domain published by the DNS drivers.

Example 3.180. Create or update DNS domain: JSON request

```
{
    "domain_entry": {
        "scope": "public",
        "project": "project1"
    }
}
```

3.33.2.2. Response

Example 3.181. Create or update DNS domain: JSON response

```
{
    "domain_entry": {
        "availability_zone": null,
        "domain": "domain1.example.org",
        "project": "project1",
        "scope": "public"
    }
}
```

3.33.3. 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

3.33.3.1. Request

This table shows the URI parameters for the delete dns domain request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{domain}	String	Registered DNS domain published by the DNS drivers.

This operation does not accept a request body.

3.33.4. 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

3.33.4.1. Request

This table shows the URI parameters for the create or update dns entry request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{domain}	String	Registered DNS domain published by the DNS drivers.
{name}	String	The name of the DNS entry.

Example 3.182. Create or update DNS entry: JSON request

```
{
  "dns_entry": {
    "ip": "192.168.53.11",
    "dns_type": "A"
  }
}
```

3.33.4.2. Response

Example 3.183. Create or update DNS entry: JSON response

```
{
  "dns_entry": {
    "domain": "domain1.example.org",
    "id": null,
    "ip": "192.168.1.1",
    "name": "instance1",
    "type": "A"
  }
}
```

3.33.5. 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

3.33.5.1. Request

This table shows the URI parameters for the find unique dns entry request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{domain}	String	Registered DNS domain published by the DNS drivers.
{name}	String	The name of the DNS entry.

This operation does not accept a request body.

3.33.5.2. Response

Example 3.184. Find unique DNS entry: JSON response

```
{
  "dns_entry": {
    "domain": "domain1.example.org",
    "id": null,
    "ip": "192.168.1.1",
    "name": "instance1",
    "type": null
  }
}
```

3.33.6. 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

3.33.6.1. Request

This table shows the URI parameters for the delete dns entry request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{domain}	String	Registered DNS domain published by the DNS drivers.
{name}	String	The name of the DNS entry.

This operation does not accept a request body.

3.33.7. 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

3.33.7.1. Request

This table shows the URI parameters for the list dns entries request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{domain}	String	Registered DNS domain published by the DNS drivers.
{ip}	String	The IP address.

This operation does not accept a request body.

3.33.7.2. Response

Example 3.185. List DNS entries: JSON response

```
{
  "dns_entries": [
    {
      "domain": "domain1.example.org",
      "id": null,
      "ip": "192.168.1.1",
      "name": "instance1",
      "type": null
    }
  ]
}
```

3.34. 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.

3.34.1. 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

3.34.1.1. Request

This table shows the URI parameters for the list floating ip pools request:

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

This operation does not accept a request body.

3.34.1.2. Response

Example 3.186. List floating IP pools: JSON response

```
{
    "floating_ip_pools": [
        {
            "name": "pool1"
        },
        {
            "name": "pool2"
        }
    ]
}
```

3.35. Floating IPs (os-floating-ips)

Assigns and allocates floating IP addresses to instances that run in an OpenStack cloud.

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	Allocates a new floating IP address to a tenant or account.
DELETE	/v2.1/{tenant_id}/os-floating-ips/{id}	Deallocates a floating IP address.

3.35.1. List floating IPs

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

3.35.1.1. Request

This table shows the URI parameters for the list floating ips request:

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

This operation does not accept a request body.

3.35.1.2. Response

Example 3.187. List floating IPs: JSON response

```
{
    "floating_ips": [
        {
            "fixed_ip": null,
            "id": 1,
            "instance_id": null,
            "ip": "10.10.10.1",
            "pool": "nova"
        },
        {
            "fixed_ip": null,
            "id": 2,
            "instance_id": null,
            "ip": "10.10.10.2",
            "pool": "nova"
        }
    ]
}
```

3.35.2. Allocate floating IP

Method	URI	Description
POST	/v2.1/{tenant_id}/os-floating-ips	Allocates a new floating IP address to a 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.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400		If there are no floating IPs available, the extension returns an error code 400 with a message indicating that no more floating IPs are available.

3.35.2.1. Request

This table shows the URI parameters for the allocate floating ip request:

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

Example 3.188. Allocate floating IP: JSON request

```
{
    "pool": "nova"
}
```

3.35.2.2. Response

Example 3.189. Allocate floating IP: JSON response

```
{
    "floating_ip": {
        "fixed_ip": null,
        "id": 1,
        "instance_id": null,
        "ip": "10.10.10.1",
        "pool": "nova"
    }
}
```

3.35.3. Deallocate floating IP

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-floating-ips/{id}	Deallocates a floating IP address.

You can use this call to deallocate and delete only manually-allocated floating IP addresses. You must manually deallocate auto-allocated floating IP addresses before you can delete them.

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

3.35.3.1. Request

This table shows the URI parameters for the deallocate floating ip request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{id}	UUID	The ID of the floating IP address.

This operation does not accept a request body.

3.36. Floating IPs bulk (os-floating-ips-bulk)

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.

3.36.1. List floating IPs

Method	URI	Description
GET	/v2.1/{tenant_id}/os-floating-ips-bulk	Lists all floating IPs.

Normal response codes: 200

3.36.1.1. Request

This table shows the URI parameters for the list floating ips request:

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

This operation does not accept a request body.

3.36.1.2. Response

Example 3.190. 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
    }
  ]
}
```

3.36.2. Create floating IPs

Method	URI	Description
POST	/v2.1/{tenant_id}/os-floating-ips-bulk	Bulk-creates floating IPs.

Normal response codes: 200

3.36.2.1. Request

This table shows the URI parameters for the create floating ips request:

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

Example 3.191. Create floating IPs: JSON request

```
{
    "floating_ips_bulk_create": {
        "ip_range": "192.168.1.0/24",
        "pool": "nova",
        "interface": "eth0"
    }
}
```

3.36.2.2. Response

Example 3.192. Create floating IPs: JSON response

```
{
    "floating_ips_bulk_create": {
        "interface": "eth0",
        "ip_range": "192.168.1.0/24",
        "pool": "nova"
    }
}
```

3.36.3. 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

3.36.3.1. Request

This table shows the URI parameters for the bulk-delete floating ips request:

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

Example 3.193. Bulk-delete floating IPs: JSON request

```
{  
    "ip_range": "192.168.1.0/24"  
}
```

3.36.3.2. Response

Example 3.194. Bulk-delete floating IPs: JSON response

```
{  
    "floating_ips_bulk_delete": "192.168.1.0/24"  
}
```

3.36.4. 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

3.36.4.1. Request

This table shows the URI parameters for the list floating ips by host request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{host_name}	String	The name of the host of interest to you.

This operation does not accept a request body.

3.36.4.2. Response

Example 3.195. 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
        }
    ]
}
```

3.37. 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/{id}	Run the fping utility to ping an instance and report whether it is alive.

3.37.1. 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

3.37.1.1. Request

This table shows the URI parameters for the ping instances request:

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

This table shows the query parameters for the ping instances request:

Name	Type	Description
all_tenants	Int <i>(Optional)</i>	Specify the <code>all_tenants=1</code> query parameter to ping instances for all tenants.
include	String <i>(Optional)</i>	Specify <code>include=uuid[,uuid...]</code> to include the instances in the results.
exclude	String <i>(Optional)</i>	Specify <code>exclude=uuid[,uuid...]</code> to exclude the instances from the results.

This operation does not accept a request body.

3.37.1.2. Response

Example 3.196. Ping instances: JSON response

```
{
  "servers": [
    {
      "alive": false,
      "id": "1d1aea35-472b-40cf-9337-8eb68480aaal",
      "project_id": "openstack"
    }
  ]
}
```

```
    ]  
}
```

3.37.2. Ping an instance

Method	URI	Description
GET	/v2.1/{tenant_id}/os-fping/{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

3.37.2.1. Request

This table shows the URI parameters for the ping an instance request:

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

This operation does not accept a request body.

3.37.2.2. Response

Example 3.197. Ping an instance: JSON response

```
{
  "server": {
    "alive": false,
    "id": "f5e6fd6d-c0a3-4f9e-aabf-d69196b6d11a",
    "project_id": "openstack"
  }
}
```

3.38. Hosts (os-hosts)

Manages physical hosts.

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.

3.38.1. List hosts

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts	Lists hosts.

Normal response codes: 200

3.38.1.1. Request

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

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

This operation does not accept a request body.

3.38.1.2. Response

Example 3.198. 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": "e73ec0bd35c64de4aladfa8b8969a1f6",
            "service": "consoleauth",
            "zone": "internal"
        },
        {
            "host_name": "396a8a0a234f476eb05fb9fbc5802ba7",
            "service": "network",
            "zone": "internal"
        },
        {
            "host_name": "abffdःda96592c4eacaf4111c28fddee17",
            "service": "scheduler",
            "zone": "internal"
        },
        {
            "host_name": "a8820f04962a4b4ba9fe2e9540c24094",
            "service": "cells",
            "zone": "internal"
        }
    ]
}
```

```
        "zone": "internal"
    }
}
```

3.38.2. 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

3.38.2.1. Request

This table shows the URI parameters for the enable host request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{host_name}	String	The name of the host.

Example 3.199. Enable host: JSON request

```
{
    "status": "enable",
    "maintenance_mode": "disable"
}
```

3.38.2.2. Response

Example 3.200. Enable host: JSON response

```
{
    "host": "65c5d5b7e3bd44308e67fc50f362aee6",
    "maintenance_mode": "off_maintenance",
    "status": "enabled"
}
```

3.38.3. Show host details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts/{host_name}	Shows details for a host.

Normal response codes: 200

3.38.3.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{host_name}	String	The name of the host.

This operation does not accept a request body.

3.38.3.2. Response

Example 3.201. 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)"
      }
    }
  ]
}
```

3.38.4. Reboot host

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/reboot	Reboots a host.

Normal response codes: 200

3.38.4.1. Request

This table shows the URI parameters for the reboot host request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{host_name}	String	The name of the host.

This operation does not accept a request body.

3.38.4.2. Response

Example 3.202. Reboot host: JSON response

```
{
    "host": "9557750dbc464741a89c907921c1cb31",
    "power_action": "reboot"
}
```

3.38.5. Shut down host

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/shutdown	Shuts down a host.

Normal response codes: 200

3.38.5.1. Request

This table shows the URI parameters for the shut down host request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{host_name}	String	The name of the host.

This operation does not accept a request body.

3.38.5.2. Response

Example 3.203. Shut down host: JSON response

```
{
    "host": "77cfa0002e4d45fe97f185968111b27b",
    "power_action": "shutdown"
}
```

3.38.6. Start host

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/startup	Starts a host.

Normal response codes: 200

3.38.6.1. Request

This table shows the URI parameters for the start host request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{host_name}	String	The name of the host.

This operation does not accept a request body.

3.38.6.2. Response

Example 3.204. Start host: JSON response

```
{
    "host": "4b392b27930343bbaa27fd5d8328a564",
    "power_action": "startup"
}
```

3.39. 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.

3.39.1. List hypervisors

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hypervisors	Lists hypervisors.

Normal response codes: 200

3.39.1.1. Request

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

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

This operation does not accept a request body.

3.39.1.2. Response

Example 3.205. List hypervisors: JSON response

```
{
    "hypervisors": [
        {
            "status": "enabled",
            "state": "up",
            "id": 1,
            "hypervisor_hostname": "fake-mini"
        }
    ]
}
```

3.39.2. 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

3.39.2.1. Request

This table shows the URI parameters for the show hypervisor statistics request:

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

This operation does not accept a request body.

3.39.2.2. Response

Example 3.206. 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
    }
}
```

3.39.3. Show hypervisor details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}	Shows details for a hypervisor.

Normal response codes: 200

3.39.3.1. Request

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

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

This operation does not accept a request body.

3.39.3.2. Response

Example 3.207. 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": "{\"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
    }
}
```

}

3.39.4. 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

3.39.4.1. Request

This table shows the URI parameters for the show hypervisor uptime request:

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

This operation does not accept a request body.

3.39.4.2. Response

Example 3.208. 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"
  }
}
```

3.40. 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.

3.40.1. 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

3.40.1.1. Request

This table shows the URI parameters for the list usage audits for an instance request:

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

This operation does not accept a request body.

3.40.1.2. Response

Example 3.209. List usage audits for an instance: JSON response

```
{
  "instance_usage_audit_logs": {
    "hosts_not_run": [
      "f4eb7cf155f4574967f8b55a7faed75"
    ],
    "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
  }
}
```

3.40.2. 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

3.40.2.1. Request

This table shows the URI parameters for the list usage audits before specified time request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID 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 <i>(Required)</i>	<p>The date and time before which to list usage audits.</p> <p>The date and time stamp format is ISO 8601:</p> <p style="background-color: #f0f0f0; padding: 2px;">CCYY-MM-DDThh:mm:ss±hh:mm</p> <p>The ±hh:mm value, if included, returns the time zone as an offset from UTC.</p> <p>For example, 2015-08-27T09:49:58-05:00.</p> <p>If you omit the time zone, the UTC time zone is assumed.</p>

This operation does not accept a request body.

3.40.2.2. Response

Example 3.210. 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
  }
}
```

3.41. Migrations (os-migrations)

Shows data on migrations.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-migrations	Lists in-progress migrations.

3.41.1. 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

3.41.1.1. Request

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

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

This operation does not accept a request body.

3.41.1.2. Response

Example 3.211. List migrations: JSON response

```
{
  "migrations": [
    {
      "created_at": "2012-10-29T13:42:02.000000",
      "dest_compute": "compute2",
      "dest_host": "1.2.3.4",
      "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"
    }
  ]
}
```

{}

3.42. 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 that are available to the project.
POST	/v2.1/{tenant_id}/os-networks/add	Adds a network to a project.
GET	/v2.1/{tenant_id}/os-networks/{id}	Shows details for a network.
DELETE	/v2.1/{tenant_id}/os-networks/{id}	Deletes a network.
POST	/v2.1/{tenant_id}/os-networks/{id}/action	Associates a network with a host.
POST	/v2.1/{tenant_id}/os-networks/{id}/action	Disassociates the host from a network.
POST	/v2.1/{tenant_id}/os-networks/{id}/action	Disassociates a network from a project so that the network can be reused.
POST	/v2.1/{tenant_id}/os-networks/{id}/action	Disassociates the project from a network.

3.42.1. 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: 202

3.42.1.1. Request

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

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

Example 3.212. 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"
  }
}
```

3.42.1.2. Response

Example 3.213. 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,
    "id": "4a1a2a3a-4a1a-4a1a-4a1a-4a1a2a3a4a1a",
    "label": "new net 111",
    "links": [
      {
        "id": "4a1a2a3a-4a1a-4a1a-4a1a-4a1a2a3a4a1a",
        "name": "new net 111"
      }
    ],
    "mac": "00:0C:29:4A:1A:2A",
    "mtu": 9000,
    "ports": [
      {
        "id": "4a1a2a3a-4a1a-4a1a-4a1a-4a1a2a3a4a1a",
        "name": "new net 111"
      }
    ],
    "subnets": [
      {
        "cidr": "10.20.105.0/24",
        "dhcp_start": "10.20.105.2",
        "gateway": "10.20.105.1",
        "id": "4a1a2a3a-4a1a-4a1a-4a1a-4a1a2a3a4a1a",
        "label": "new net 111"
      }
    ]
  }
}
```

```
        "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
    }
}
```

3.42.2. List networks

Method	URI	Description
GET	/v2.1/{tenant_id}/os-networks	Lists networks that are available to the project.

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

3.42.2.1. Request

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

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

This operation does not accept a request body.

3.42.2.2. Response

Example 3.214. 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
    }
]
```

3.42.3. 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 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

3.42.3.1. Request

This table shows the URI parameters for the add network request:

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

Example 3.215. Add network: JSON request

```
{
    "id": "1"
}
```

3.42.4. Show network details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-networks/{id}	Shows details for 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

3.42.4.1. Request

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

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

This operation does not accept a request body.

3.42.4.2. Response

Example 3.216. 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
    }
}
```

3.42.5. Delete network

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-networks/{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

3.42.5.1. Request

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

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

This operation does not accept a request body.

3.42.6. Associate host

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks/{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

3.42.6.1. Request

This table shows the URI parameters for the associate host request:

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

Example 3.217. Associate host: JSON request

```
{
    "associate_host": "testHost"
}
```

3.42.7. Disassociate host

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks/{id}/action	Disassociates the 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

3.42.7.1. Request

This table shows the URI parameters for the disassociate host request:

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

Example 3.218. Disassociate host: JSON request

```
{  
    "disassociate_host": null  
}
```

3.42.8. Disassociate network

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks/{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

3.42.8.1. Request

This table shows the URI parameters for the disassociate network request:

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

Example 3.219. Disassociate network: JSON request

```
{  
    "disassociate": null  
}
```

3.42.9. Disassociate project

Method	URI	Description
POST	/v2.1/{tenant_id}/os-networks/{id}/action	Disassociates the 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

3.42.9.1. Request

This table shows the URI parameters for the disassociate project request:

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

Example 3.220. Disassociate project: JSON request

```
{
    "disassociate_project": null
}
```

3.43. 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.

3.43.1. 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

3.43.1.1. Request

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

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

This operation does not accept a request body.

3.43.1.2. Response

Example 3.221. 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
    }
}
```

3.43.2. 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

3.43.2.1. Request

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

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

Example 3.222. 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
    }
}
```

3.43.2.2. Response

Example 3.223. 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
    }
}
```

3.44. Quota sets (os-quota-sets)

Permits administrators, depending on policy settings, to view quotas for a project and view and update default quotas.

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-quota-sets	Deletes a quota for tenant.
PUT	/v2.1/{tenant_id}/os-quota-sets	Force-updates quota for tenant.
PUT	/v2.1/{tenant_id}/os-quota-sets	Updates quota for tenant.
GET	/v2.1/{tenant_id}/os-quota-sets/defaults	Shows default quotas for tenant.
GET	/v2.1/{tenant_id}/os-quota-sets/detail	Lists quotas with details for a tenant.
PUT	/v2.1/{tenant_id}/os-quota-sets/ {? user_id}	Updates quota for user.
DELETE	/v2.1/{tenant_id}/os-quota-sets/ {? user_id}	Deletes quota for a user.

3.44.1. Delete quota for tenant

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-quota-sets	Deletes a quota for tenant.

Normal response codes: 204

3.44.1.1. Request

This table shows the URI parameters for the delete quota for tenant request:

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

This operation does not accept a request body.

3.44.2. Force-update quota

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-quota-sets	Force-updates quota for tenant.

Normal response codes: 200

3.44.2.1. Request

This table shows the URI parameters for the force-update quota request:

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

Example 3.224. Force-update quota: JSON request

```
{
    "quota_set": {
        "force": "True",
        "instances": 45
    }
}
```

3.44.2.2. Response

Example 3.225. Force-update quota: JSON response

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

3.44.3. Update quota

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-quota-sets	Updates quota for tenant.

Normal response codes: 200

3.44.3.1. Request

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

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

Example 3.226. Update quota: JSON request

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

3.44.3.2. Response

Example 3.227. Update quota: JSON response

```
{
    "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,
        "server_groups": 10,
        "server_group_members": 10
    }
}
```

3.44.4. Show default quotas

Method	URI	Description
GET	/v2.1/{tenant_id}/os-quota-sets/defaults	Shows default quotas for tenant.

Normal response codes: 200

3.44.4.1. Request

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

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

This operation does not accept a request body.

3.44.4.2. Response

Example 3.228. 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,
        "server_groups": 10,
        "server_group_members": 10
    }
}
```

3.44.5. Show quotas (detailed)

Method	URI	Description
GET	/v2.1/{tenant_id}/os-quota-sets/detail	Lists quotas with details for a tenant.

Normal response codes: 200

3.44.5.1. Request

This table shows the URI parameters for the show quotas (detailed) request:

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

This operation does not accept a request body.

3.44.5.2. Response

Example 3.229. Show quotas (detailed): 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,
        "server_groups": 10,
        "server_group_members": 10
    }
}
```

3.44.6. Update quota for user

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-quota-sets/{?user_id}	Updates quota for user.

Normal response codes: 200

3.44.6.1. Request

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

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

Example 3.230. Update quota for user: JSON request

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

3.44.6.2. Response

Example 3.231. Update quota for user: JSON response

```
{
    "quota_set": {
        "cores": 20,
        "fixed_ips": -1,
        "floating_ips": 10,
        "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,
        "server_groups": 10,
        "server_group_members": 10
    }
}
```

3.44.7. Delete quota for user

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-quota-sets/ {? user_id}	Deletes quota for a user.

Normal response codes: 204

3.44.7.1. Request

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

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

This operation does not accept a request body.

3.45. 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.
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.

3.45.1. List security groups

Method	URI	Description
GET	/v2.1/{tenant_id}/os-security-groups	Lists security groups.

Normal response codes: 200

3.45.1.1. Request

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

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

This operation does not accept a request body.

3.45.1.2. Response

Example 3.232. List security groups: JSON response

```
{
    "security_groups": [
        {
            "description": "default",
            "id": "3fb26eb3-581b-4420-9963-b0879a026506",
            "name": "default",
            "rules": [],
            "tenant_id": "openstack"
        }
    ]
}
```

3.45.2. Create security group

Method	URI	Description
POST	/v2.1/{tenant_id}/os-security-groups	Creates a security group.

Normal response codes: 200

3.45.2.1. Request

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

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

Example 3.233. Create security group: JSON request

```
{
    "security_group": {
        "name": "test",
        "description": "test"
    }
}
```

3.45.2.2. Response

Example 3.234. Create security group: JSON response

```
{
    "security_group": {
        "description": "test",
        "id": "de3ff110-46b2-4ace-bb76-a2e1ed741b95",
        "name": "test",
        "rules": [],
        "tenant_id": "openstack"
    }
}
```

3.45.3. 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

3.45.3.1. Request

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

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

This operation does not accept a request body.

3.45.3.2. Response

Example 3.235. Show security group: JSON response

```
{
    "security_group": {
        "description": "default",
        "id": "de3ff110-46b2-4ace-bb76-a2e1ed741b95",
        "name": "default",
        "rules": [],
        "tenant_id": "openstack"
    }
}
```

3.45.4. 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

3.45.4.1. Request

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

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

Example 3.236. Update security group: JSON request

```
{
  "security_group": {
    "name": "mysecgroup",
    "description": "my security group"
  }
}
```

3.45.4.2. Response

Example 3.237. Update security group: JSON response

```
{
  "security_group": {
    "rules": [],
    "tenant_id": "a52cdb9cc7854a39a23d3af73a40899e",
    "id": "01fbade5-b664-42f6-83ae-4e214f4263fa",
    "name": "mysecgroup",
    "description": "my security group"
  }
}
```

3.45.5. 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

3.45.5.1. Request

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

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

This operation does not accept a request body.

3.45.6. 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

3.45.6.1. Request

This table shows the URI parameters for the list security groups by server request:

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server.

This operation does not accept a request body.

3.45.6.2. Response

Example 3.238. List security groups by server: JSON response

```
{
    "security_groups": [
        {
            "description": "default",
            "id": "3fb26eb3-581b-4420-9963-b0879a026506",
            "name": "default",
            "rules": [],
            "tenant_id": "openstack"
        }
    ]
}
```

3.46. 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.

3.46.1. 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)

3.46.1.1. Request

This table shows the URI parameters for the list default security group rules request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

3.46.1.2. Response

Example 3.239. 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
        }
    ]
}
```

3.46.2. Create default security group rule

Method	URI	Description
POST	/v2.1/{tenant_id}/os-security-group-default-rules	Creates a default security group rule.

You must specify an IP protocol (`ip_protocol`) value if you specify a source (`from_port`) or destination (`to_port`) port value. Otherwise, the operation returns the Bad request (400) Source/destination port requires a protocol error.

Normal response codes: 200

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

3.46.2.1. Request

This table shows the URI parameters for the create default security group rule request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.

Example 3.240. 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"
  }
}
```

3.46.2.2. Response

Example 3.241. 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
  }
}
```

3.46.3. 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)

3.46.3.1. Request

This table shows the URI parameters for the show default security group rule details request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{security_group_default_rule_id}	UUID	The unique identifier of the security group rule.

This operation does not accept a request body.

3.46.3.2. Response

Example 3.242. 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
    }
}
```

3.46.4. 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)

3.46.4.1. Request

This table shows the URI parameters for the delete default security group rule request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{security_group_default_rule_id}	UUID	The unique identifier of the security group rule.

This operation does not accept a request body.

3.47. 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.

3.47.1. 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

3.47.1.1. Request

This table shows the URI parameters for the create security group rule request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant.

Example 3.243. 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"
    }
}
```

3.47.1.2. Response

Example 3.244. Create security group rule: JSON response

```
{
    "security_group_rule": {
        "id": "2d021cf1-ce4b-4292-994f-7a785d62a144",
        "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
    }
}
```

3.47.2. 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

3.47.2.1. Request

This table shows the URI parameters for the delete security group rule request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant.
{security_group_rule_id}	Uuid	The ID of the security group rule.

This operation does not accept a request body.

3.48. Execute external events (os-server-external-events)

Executes external events.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-server-external-events	Creates events.

3.48.1. Create events

Method	URI	Description
POST	/v2.1/{tenant_id}/os-server-external-events	Creates events.

Normal response codes: 200

3.48.1.1. Request

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

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

Example 3.245. Create events: JSON request

```
{
  "events": [
    {
      "name": "test-event",
      "tag": "foo",
      "status": "completed",
      "server_uuid": "3df201cf-2451-44f2-8d25-a4ca826fc1f3"
    }
  ]
}
```

3.48.1.2. Response

Example 3.246. Create events: JSON response

```
{
  "events": [
    {
      "code": 200,
      "name": "network-changed",
      "server_uuid": "ff1df7b2-6772-45fd-9326-c0a3b05591c2",
      "status": "completed",
      "tag": "foo"
    }
  ]
}
```

3.49. 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.
POST	/v2.1/{tenant_id}/os-server-groups	Creates a server group.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-server-groups/{ServerGroup_id}	Shows details for a server group.
DELETE	/v2.1/{tenant_id}/os-server-groups/{ServerGroup_id}	Deletes a server group.

3.49.1. 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

3.49.1.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID 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.

3.49.1.2. Response

Example 3.247. List server groups: JSON response

```
{
    "server_groups": [
        {
            "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
            "name": "test",
            "policies": [
                "anti-affinity"
            ],
            "members": [],
            "metadata": {}
        }
    ]
}
```

3.49.2. Create server group

Method	URI	Description
POST	/v2.1/{tenant_id}/os-server-groups	Creates a server group.

Normal response codes: 200

3.49.2.1. Request

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

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

Example 3.248. Create server group: JSON request

```
{
    "server_group": {
        "name": "test",
        "policies": [
            "anti-affinity"
        ]
    }
}
```

3.49.2.2. Response

Example 3.249. Create server group: JSON response

```
{
    "server_group": {
        "id": "5bbcc3c4-1da2-4437-a48a-66f15b1b13f9",
        "name": "test",
        "policies": [
            "anti-affinity"
        ],
        "members": [],
        "metadata": {}
    }
}
```

3.49.3. Show server group details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-server-groups/{ServerGroup_id}	Shows details for a server group.

Normal response codes: 200

3.49.3.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{ServerGroup_id}	String	The server group ID.

This operation does not accept a request body.

3.49.3.2. Response

Example 3.250. Show server group details: JSON response

```
{
    "server_group": {
        "id": "5bcc3c4-1da2-4437-a48a-66f15b1b13f9",
        "name": "test",
        "policies": [
            "anti-affinity"
        ],
        "members": [],
        "metadata": {}
    }
}
```

3.49.4. Delete server group

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-server-groups/{ServerGroup_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)

3.49.4.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{ServerGroup_id}	String	The server group ID.

This operation does not accept a request body.

3.50. Manage services (os-services)

Lists all or only disabled Compute services for all hosts in a tenant. Logs information for disabled services. Enables or disables scheduling for, forces down, unsets the forced-down state of, or deletes a service.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-services	Lists all services for a tenant. Includes reasons, if available, for why any disabled services were disabled.
GET	/v2.1/{tenant_id}/os-services/detail	Lists disabled services for a tenant. Includes reasons, if available, for why disabled services were disabled.
PUT	/v2.1/{tenant_id}/os-services/enable	Enables scheduling for a service.
PUT	/v2.1/{tenant_id}/os-services/disable	Disables scheduling for a service.
PUT	/v2.1/{tenant_id}/os-services/disable-log-reason	Logs information to the service table about why a service was disabled.
PUT	/v2.1/{tenant_id}/os-services/force-down	Forces down a service or unsets the forced-down state of a service.
DELETE	/v2.1/{tenant_id}/os-services	Deletes a service.

3.50.1. List services

Method	URI	Description
GET	/v2.1/{tenant_id}/os-services	Lists all services for a tenant. Includes reasons, if available, for why any disabled services were disabled.

Normal response codes: 200

3.50.1.1. Request

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

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

This operation does not accept a request body.

3.50.1.2. Response

Example 3.251. List services: JSON response

```
{
  "services": [
    {
      "id": 1,
      "binary": "nova-scheduler",
      "disabled_reason": "test1",
      "host": "host1",
      "state": "up",
      "forced_down": false,
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:02.000000",
      "zone": "internal"
    },
    {
      "id": 2,
      "binary": "nova-compute",
      "disabled_reason": "test2",
      "host": "host1",
      "state": "up",
      "forced_down": false,
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:05.000000",
      "zone": "nova"
    },
    {
      "id": 3,
      "binary": "nova-scheduler",
      "disabled_reason": null,
      "host": "host2",
      "state": "down",
      "forced_down": false,
      "status": "enabled",
      "updated_at": "2012-09-19T06:55:34.000000",
      "zone": "internal"
    }
  ]
}
```

```
        },
        {
            "id": 4,
            "binary": "nova-compute",
            "disabled_reason": "test4",
            "host": "host2",
            "state": "down",
            "forced_down": false,
            "status": "disabled",
            "updated_at": "2012-09-18T08:03:38.000000",
            "zone": "nova"
        }
    ]
}
```

3.50.2. List disabled services

Method	URI	Description
GET	/v2.1/{tenant_id}/os-services/detail	Lists disabled services for a tenant. Includes reasons, if available, for why disabled services were disabled.

Normal response codes: 200

3.50.2.1. Request

This table shows the URI parameters for the list disabled services request:

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

This operation does not accept a request body.

3.50.2.2. Response

Example 3.252. List disabled services: JSON response

```
{
    "services": [
        {
            "id": 1,
            "binary": "nova-scheduler",
            "disabled_reason": "test1",
            "host": "host1",
            "state": "up",
            "forced_down": false,
            "status": "disabled",
            "updated_at": "2012-10-29T13:42:02.000000",
            "zone": "internal"
        },
        {
            "id": 2,
            "binary": "nova-compute",
            "disabled_reason": "test2",
            "host": "host1",
            "state": "up",
            "forced_down": false,
            "status": "disabled",
            "updated_at": "2012-10-29T13:42:05.000000",
            "zone": "nova"
        },
        {
            "id": 3,
            "binary": "nova-scheduler",
            "disabled_reason": null,
            "host": "host2",
            "state": "down",
            "forced_down": false,
            "status": "enabled",
            "updated_at": "2012-09-19T06:55:34.000000",
            "zone": "internal"
        }
    ]
}
```

```
        },
        {
            "id": 4,
            "binary": "nova-compute",
            "disabled_reason": "test4",
            "host": "host2",
            "state": "down",
            "forced_down": false,
            "status": "disabled",
            "updated_at": "2012-09-18T08:03:38.000000",
            "zone": "nova"
        }
    ]
}
```

3.50.3. Enable scheduling for a service

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-services/enable	Enables scheduling for a service.

Specify the service by its host name and binary name.

Normal response codes: 200

3.50.3.1. Request

This table shows the URI parameters for the enable scheduling for a service request:

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

Example 3.253. Enable scheduling for a service: JSON request

```
{
    "host": "host1",
    "binary": "nova-compute"
}
```

3.50.3.2. Response

Example 3.254. Enable scheduling for a service: JSON response

```
{
    "service": {
        "binary": "nova-compute",
        "host": "host1",
        "status": "enabled"
    }
}
```

3.50.4. Disable scheduling for a service

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-services/disable	Disables scheduling for a service.

Specify the service by its host name and binary name.

Normal response codes: 200

3.50.4.1. Request

This table shows the URI parameters for the disable scheduling for a service request:

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

Example 3.255. Disable scheduling for a service: JSON request

```
{
    "host": "host1",
    "binary": "nova-compute"
}
```

3.50.4.2. Response

Example 3.256. Disable scheduling for a service: JSON response

```
{
    "service": {
        "binary": "nova-compute",
        "host": "host1",
        "status": "disabled"
    }
}
```

3.50.5. Log disabled service information

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-services/disable-log-reason	Logs information to the service table about why a service was disabled.

Specify the service by its host name and binary name.

Normal response codes: 200

3.50.5.1. Request

This table shows the URI parameters for the log disabled service information request:

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

Example 3.257. Log disabled service information: JSON request

```
{
    "host": "host1",
    "binary": "nova-compute",
    "disabled_reason": "test2"
}
```

3.50.5.2. Response

Example 3.258. Log disabled service information: JSON response

```
{
    "service": {
        "binary": "nova-compute",
        "disabled_reason": "test2",
        "host": "host1",
        "status": "disabled"
    }
}
```

3.50.6. Force down service

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-services/force-down	Forces down a service or unsets the forced-down state of a service.

Specify the service by its host name and binary name.

Normal response codes: 200

3.50.6.1. Request

This table shows the URI parameters for the force down service request:

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

Example 3.259. Force down service: JSON request

```
{
    "host": "host1",
    "binary": "nova-compute",
    "forced_down": true
}
```

3.50.6.2. Response

Example 3.260. Force down service: JSON response

```
{
    "service": {
        "host": "host1",
        "binary": "nova-compute",
        "forced_down": true
    }
}
```

3.50.7. Delete service

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-services	Deletes a service.

Specify the service by its host name and binary name.

Normal response codes: 204

Error response codes: 404

3.50.7.1. Request

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

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

Example 3.261. Delete service: JSON request

```
{
    "host": "host1",
    "binary": "nova-compute"
}
```

3.51. Usage reports (os-simple-tenant-usage)

Reports usage statistics on compute and storage resources.

Method	URI	Description
GET	/v2.1/os-simple-tenant-usage	Lists usage information for all tenants.
GET	/v2.1/os-simple-tenant-usage/{tenant_id}	Shows usage details for a tenant.

3.51.1. List tenant usage for all tenants

Method	URI	Description
GET	/v2.1/os-simple-tenant-usage	Lists usage information for all tenants.

Normal response codes: 200

3.51.1.1. Response

Example 3.262. 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  
        }  
    ]  
}
```

3.51.2. Show usage details for tenant

Method	URI	Description
GET	/v2.1/os-simple-tenant-usage/{tenant_id}	Shows usage details for a tenant.

Normal response codes: 200

3.51.2.1. Request

This table shows the URI parameters for the show usage details for tenant request:

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

3.51.2.2. Response

Example 3.263. 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
    }
}
```

3.52. 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 project networks.
GET	/v2.1/{tenant_id}/os-tenant-networks/{id}	Shows details for a project network.
DELETE	/v2.1/{tenant_id}/os-tenant-networks/{id}	Deletes a project network.

3.52.1. 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: 202

3.52.1.1. Request

This table shows the URI parameters for the create project network request:

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

Example 3.264. Create project network: JSON request

```
{
  "network": {
    "label": "public",
    "cidr": "172.0.0.0/24",
    "vlan_start": 1,
    "num_networks": 1,
    "network_size": 255
  }
}
```

3.52.1.2. Response

Example 3.265. Create project network: JSON response

```
{
  "network": {
    "cidr": "172.0.0.0/24",
    "id": "5bcc3c4-1da2-4437-a48a-66f15b1b13f9",
    "label": "public"
  }
}
```

3.52.2. List project networks

Method	URI	Description
GET	/v2.1/{tenant_id}/os-tenant-networks	Lists 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

3.52.2.1. Request

This table shows the URI parameters for the list project networks request:

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

This operation does not accept a request body.

3.52.2.2. Response

Example 3.266. List project networks: 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"
    }
  ]
}
```

3.52.3. Show project network details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-tenant-networks/{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

3.52.3.1. Request

This table shows the URI parameters for the show project network details request:

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

This operation does not accept a request body.

3.52.3.2. Response

Example 3.267. Show project network details: JSON response

```
{
  "network": {
    "cidr": "172.0.0.0/24",
    "id": "5bcc3c4-1da2-4437-a48a-66f15b1b13f9",
    "label": "public"
  }
}
```

3.52.4. Delete project network

Method	URI	Description
DELETE	/v2.1/{tenant_id}/os-tenant-networks/{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

3.52.4.1. Request

This table shows the URI parameters for the delete project network request:

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

This operation does not accept a request body.

3.53. 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.
DELETE	/v2.1/{tenant_id}/os-snapshots/{snapshot_id}	Deletes a snapshot from the account.

3.53.1. 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)

3.53.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

3.53.1.2. Response

Example 3.268. 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"
    }
  ]
}
```

3.53.2. Create volume

Method	URI	Description
POST	/v2.1/{tenant_id}/os-volumes	Creates a volume.

Normal response codes: 201

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

3.53.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.

Example 3.269. 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"
  }
}
```

3.53.2.2. Response

Example 3.270. 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"
  }
}
```

3.53.3. 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)

3.53.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

3.53.3.2. Response

Example 3.271. List volumes with details: 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"
    }
  ]
}
```

3.53.4. 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)

3.53.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{volume_id}	String	The unique identifier for a volume.

This operation does not accept a request body.

3.53.4.2. Response

Example 3.272. 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"
    }
}
```

3.53.5. 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)

3.53.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{volume_id}	String	The unique identifier for a volume.

This operation does not accept a request body.

3.53.5.2. Response

Example 3.273. 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.

3.53.6. 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)

3.53.6.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

3.53.6.2. Response

Example 3.274. 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": {}
        }
    ]
}
```

3.53.7. 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)

3.53.7.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{volume_type_id}	String	The unique identifier for a volume type.

This operation does not accept a request body.

3.53.7.2. Response

Example 3.275. Show volume type details: JSON response

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

3.53.8. Create snapshot

Method	URI	Description
POST	/v2.1/{tenant_id}/os-snapshots	Creates a snapshot.

Normal response codes: 201

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

3.53.8.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.

Example 3.276. Create snapshot: JSON request

```
{
  "snapshot": {
    "createdAt": "2013-02-25T16:27:54.680544",
    "displayDescription": "Daily backup",
    "displayName": "snap-001",
    "id": 100,
    "size": 100,
    "status": "available",
    "volumeId": "521752a6-acf6-4b2d-bc7a-119f9148cd8c"
  }
}
```

3.53.8.2. Response

Example 3.277. 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
  }
}
```

3.53.9. 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)

3.53.9.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

3.53.9.2. Response

Example 3.278. 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
    }
  ]
}
```

3.53.10. 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)

3.53.10.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.

This operation does not accept a request body.

3.53.10.2. Response

Example 3.279. 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
    }
  ]
}
```


3.53.11. 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)

3.53.11.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{snapshot_id}	String	The unique identifier for a snapshot.

This operation does not accept a request body.

3.53.11.2. Response

Example 3.280. 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
  }
}
```

3.53.12. 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)

3.53.12.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{snapshot_id}	String	The unique identifier for a snapshot.

This operation does not accept a request body.

3.53.12.2. Response

Example 3.281. 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.

4. 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.
DELETE	/v1.0/{accountId}/instances/{instanceId}	Deletes a database instance, including any associated data.
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.
GET	/v1.0/{accountId}/instances/{instanceId}	Lists the configuration defaults for an 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.
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.
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, along with the associated databases for that user.
DELETE	/v1.0/{accountId}/instances/{instanceId}/users/{name}	Deletes a user for a database instance.
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.
Datastores (datastores)		

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.
Configuration groups (configurations)		
POST	/v1.0/{accountId}/configurations	Creates a configuration group.
GET	/v1.0/{accountId}/configurations	Lists all configuration groups.

4.1. API versions

Method	URI	Description
GET	/	Lists information about all Database Service API versions.
GET	/v1.0	Shows details for the Database Service API v1.0.

4.1.1. 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)

4.1.1.1. Request

Example 4.1. 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.

4.1.1.2. Response

Example 4.2. 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"
    }
  ]
}
```

4.1.2. 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)

4.1.2.1. Request

Example 4.3. 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.

4.1.2.2. Response

Example 4.4. 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"
                }
            ]
        }
    ]
}
```

4.2. Database instances (instances)

Method	URI	Description
POST	/v1.0/{accountId}/instances	Creates a database instance.

Method	URI	Description
GET	/v1.0/{accountId}/instances	Lists information, including status, for all database instances.
GET	/v1.0/{accountId}/instances/{instanceId}	Shows database instance details.
DELETE	/v1.0/{accountId}/instances/{instanceId}	Deletes a database instance, including any associated data.
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.
GET	/v1.0/{accountId}/instances/{instanceId}	Lists the configuration defaults for an 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.

4.2.1. Create database instance

Method	URI	Description
POST	/v1.0/{accountId}/instances	Creates a database instance.

Asynchronously provisions a new 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, and users associated to those databases.
- 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)

4.2.1.1. 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 4.5. 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",
    "volume": {
      "size": 10
    }
  }
}
```

```

"users": [
    {
        "databases": [
            {
                "name": "sampledb"
            }
        ],
        "name": "demouser",
        "password": "secretsecret"
    }
],
"volume": {
    "size": 2
}
}
}

```

4.2.1.2. Response

Example 4.6. 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.rackspaceclouddb.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": "2012-01-25T21:53:09Z"
    }
}

```

```
        "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. There are two kinds of link relations associated with resources. A `self` link contains a *versioned* link to the resource. These links should be used 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.

4.2.2. 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)

4.2.2.1. 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 4.7. 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.

4.2.2.2. Response

Example 4.8. 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"
                    }
                ]
            }
        }
    ]
}
```

```
        "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
}
}
]
}
```

4.2.3. 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 used attribute is *not* returned in the response when the instance status is BUILD, REBOOT, RESIZE, or ERROR.

The list operations return a DNS-resolvable host name that is associated with 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)

4.2.3.1. 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 instance ID for the database instance.

Example 4.9. 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.

4.2.3.2. Response

Example 4.10. 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.rackspaceclouddb.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
        }
    }
}
```

4.2.4. 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

This operation is not permitted when the instance state is either REBUILDING 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)

4.2.4.1. 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 instance ID for the database instance.

Example 4.11. 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.

4.2.5. 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)

4.2.5.1. 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 instance ID for the database instance.

Example 4.12. 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"
    }
}
```

4.2.5.2. Response

Example 4.13. 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.

4.2.6. 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)

4.2.6.1. 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 instance ID for the database instance.

Example 4.14. 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": {}
}
```

4.2.6.2. Response

Example 4.15. 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.

4.2.7. List configuration defaults

Method	URI	Description
GET	/v1.0/{accountId}/instances/{instanceId}	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)

4.2.7.1. 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 instance ID for the database instance.

Example 4.16. 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.

4.2.7.2. Response

Example 4.17. 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"
  }
}
}
```

4.2.8. 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 enables login from any host for the root user and provides the user with a generated root password.



Note

Changes that you make as a root user can have detrimental effects on the database instance and unpredictable behavior for API operations. When you enable the root user, you accept the possibility that we will not be able to support your database instance. We may 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)

4.2.8.1. 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 instance ID for the database instance.

Example 4.18. 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.

4.2.8.2. Response

Example 4.19. 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"  
    }  
}
```

4.2.9. 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)

4.2.9.1. 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 instance ID for the database instance.

Example 4.20. 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.

4.2.9.2. Response

Example 4.21. 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
}
```

4.3. Database instance actions (action)

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/action	Resizes the memory for an instance.

Method	URI	Description
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.

4.3.1. 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)

4.3.1.1. 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 instance ID for the database instance.

Example 4.22. 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"
    }
}
```

4.3.1.2. Response

Example 4.23. 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.

4.3.2. 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)

4.3.2.1. 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 instance ID for the database instance.

Example 4.24. 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
        }
    }
}
```

4.3.2.2. Response

Example 4.25. 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.

4.3.3. 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 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)

4.3.3.1. 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 instance ID for the database instance.

Example 4.26. 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": {}
}
```

4.3.3.2. Response

Example 4.27. 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.

4.4. Databases (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.

4.4.1. Create database

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/databases	Creates a database within an instance.

This operation creates a database within an instance.

The name of the database is a required attribute.

You can specify these additional attributes for each database: `collate` and `character_set`.

Table 4.1. Required and optional attributes for creating a database

Name	Description	Required
name	Specifies the database name for creating the database. See the request examples for the required xml/json format.	Yes
character_set	Set of symbols and encodings. The default character set is <code>utf8</code> .	No
collate	Set of rules for comparing characters in a character set. The default value for collate is <code>utf8_general_ci</code> .	No

For information about supported character sets and collations, see the MySQL documentation at [Character Sets and Collations in MySQL](#).



Note

The `lost+found`, `information_schema`, and `mysql` database names are reserved and cannot be used to create databases.

See the following tables for information about characters that are valid for creating database names.

Table 4.2. Valid characters in a database name

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
'_' is allowed anywhere in the database name

Table 4.3. Characters that *cannot* be used in a database name

Character
Single quotes
Double quotes
Back quotes
Semicolons
Commas
Back slashes
Forward slashes

Table 4.4. Length restrictions for database name

Restriction	Value
Database-name maximum length	64

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)

4.4.1.1. 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 instance ID for the database instance.

Example 4.28. 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"
    }
  ]
}
```

4.4.1.2. Response

Example 4.29. 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.

4.4.2. 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, not the system databases. The system databases (mysql, information_schema, lost+found) can only be viewed by a database administrator.

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)

4.4.2.1. 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 instance ID for the database instance.

Example 4.30. 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.

4.4.2.2. Response

Example 4.31. 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": "anotherexampledb"
    },
    {
      "name": "anotherexampledb"
    }
  ]
}
```

```
        "name": "exampledbs"
    },
    {
        "name": "nextround"
    },
    {
        "name": "sampledb"
    },
    {
        "name": "testingdb"
    }
]
```

4.4.3. Delete database

Method	URI	Description
DELETE	/v1.0/{accountId}/instances/{instanceId}/databases/{databaseName}	Deletes a database.

Note that all data associated with the database is also 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)

4.4.3.1. 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 instance ID for the database instance.
{databaseName}	String	The name for the database.

Example 4.32. 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.

4.4.3.2. Response

Example 4.33. 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.

4.5. Users (users)

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/users	Creates a user for a database instance.

Method	URI	Description
GET	/v1.0/{accountId}/instances/{instanceId}/users	Lists the users in a database instance, along with the associated databases for that user.
DELETE	/v1.0/{accountId}/instances/{instanceId}/users/{name}	Deletes a user for a database instance.

4.5.1. 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 based on the configuration defined in the request object. After the request is validated and progress begins on the provisioning process, a 202 Accepted response object is returned.

If the corresponding request cannot be fulfilled due to insufficient or invalid data, an HTTP 400 "Bad Request" error response is returned with information regarding the nature of the failure. Failures in the validation process are non-recoverable and require the caller to correct the cause of the failure and POST the request again.

The following table lists the required attributes for creating user. See the request examples for the required xml/json format:

Table 4.5. Required attributes for creating a 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

- A user is granted all privileges on the specified databases.
- The following user name is reserved and cannot be used for creating users: root.

See the following tables for information about characters that are valid/invalid for creating database names, user names, and passwords.

Table 4.6. Valid characters in a 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 4.7. Characters that *cannot* be used in a 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 4.8. 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)

4.5.1.1. 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 instance ID for the database instance.

Example 4.34. 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"
}
]
```

4.5.1.2. Response

Example 4.35. 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.

4.5.2. List database instance users

Method	URI	Description
GET	/v1.0/{accountId}/instances/{instanceId}/users	Lists the users in a database instance, along with the associated databases for that user.



Note

This operation does not return the system users (database administrators that administer the health of the database). Also, this operation returns the "root" user only if "root" user 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)

4.5.2.1. 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 instance ID for the database instance.

Example 4.36. 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.

4.5.2.2. Response

Example 4.37. 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"
        }
    ]
}
```

4.5.3. Delete user

Method	URI	Description
DELETE	/v1.0/{accountId}/instances/{instanceId}/users/{name}	Deletes a user for a database instance.



Warning

There is a bug in a Python library that Rackspace is using that can cause incorrect user deletions to occur if a period (.) is used in the user name. In this case, the user name is truncated to remove the portion of the name from the period to the end, leaving only the portion from the beginning up to the period. For example, for a user named "my.userA", the bug would truncate the user name to "my", and if the user "my" exists, that user will be incorrectly deleted. To avoid the problem, do not use periods in user names.

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)

4.5.3.1. 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 instance ID for the database instance.
{name}	String	The name for the user.

Example 4.38. 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.

4.5.3.2. Response

Example 4.39. 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.

4.6. Flavors (flavors)

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.

4.6.1. 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)

4.6.1.1. 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 4.40. 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.

4.6.1.2. Response

Example 4.41. 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
}
]
```

4.6.2. 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 should be an integer. If a floating point value is used for the flavorId parameter, the decimal portion is truncated and the integer portion is used as the value of the flavorId.

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)

4.6.2.1. 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 flavor ID for the flavor.

Example 4.42. 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.

4.6.2.2. Response

Example 4.43. 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
    }
}
```

4.7. Datastores (datastores)

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.

4.7.1. 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)

4.7.1.1. 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}	Uuid	The name of the data store whose versions you want to list.

Example 4.44. 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.

4.7.1.2. Response

Example 4.45. 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"  
}
```

4.7.2. 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)

4.7.2.1. 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 ID of the data store version.

Example 4.46. 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.

4.7.2.2. Response

Example 4.47. 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": "f597f709-70ef-474d-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": "f597f709-70ef-474d-ac18-2c6abd35a758"
    },
    {
        "name": "connect_timeout",
```

```
        "min": 2,
        "max": 31536000,
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef474d-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": "f597f709-70ef474d-ac18-2c6abd35a758"
    },
    {
        "name": "server_id",
        "min": 0,
        "max": 4294967295,
```

```
        "restart_required": false,
        "type": "integer",
        "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
    }
]
```

4.7.3. 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)

4.7.3.1. 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 ID of the data store version.
{parameter_name}	Uuid	Name of the parameter whose details you want.

Example 4.48. 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.

4.7.3.2. Response

Example 4.49. 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"
}
```

4.8. Configuration groups (configurations)

Method	URI	Description
POST	/v1.0/{accountId}/configurations	Creates a configuration group.
GET	/v1.0/{accountId}/configurations	Lists all configuration groups.

4.8.1. 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)

4.8.1.1. 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 4.50. 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"
  }
}
```

4.8.1.2. Response

Example 4.51. 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
}
```

4.8.2. 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)

4.8.2.1. 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 4.52. 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.

4.8.2.2. Response

Example 4.53. 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-efff-4c2b-8cff-bcd949c518a5",
      "description": null
    }
  ]
}
```

```
        }  
    ]  
}
```

5. 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 image registry.
GET	/v1.1/{tenant_id}/images/{image_id}	Shows details about an image.
POST	/v1.1/{tenant_id}/images/{image_id}	Registers an image in the image registry.
DELETE	/v1.1/{tenant_id}/images/{image_id}	Removes an image from the image 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 the cluster template.
PUT	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Updates the cluster template.
DELETE	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Deletes the 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 cluster by ID.

Method	URI	Description
DELETE	/v1.1/{tenant_id}/clusters/{cluster_id}	Deletes a cluster.
PUT	/v1.1/{tenant_id}/clusters/{cluster_id}	Scales a cluster.
PATCH	/v1.1/{tenant_id}/clusters/{cluster_id}	Updates 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.
DELETE	/v1.1/{tenant_id}/data-sources/{data_source_id}	Deletes a data source.
PUT	/v1.1/{tenant_id}/data-sources/{data_source_id}	Updates 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	Gets data for a job binary internal.
Job binaries		
GET	/v1.1/{tenant_id}/job-binaries	Lists the available job binaries.
POST	/v1.1/{tenant_id}/job-binaries	Creates a job binary.
GET	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}	Shows details for a job binary.
DELETE	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}	Deletes a job binary.
PUT	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}	Updates a job binary.
GET	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}/data	Gets 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 job types.

5.1. 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.

5.1.1. List plugins

Method	URI	Description
GET	/v1.1/{tenant_id}/plugins	Lists all registered plugins.

Normal response codes: 200

5.1.1.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

This operation does not accept a request body.

5.1.1.2. Response

Example 5.1. 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"
    }
]
```

5.1.2. Show plugin details

Method	URI	Description
GET	/v1.1/{tenant_id}/plugins/{plugin_name}	Shows details for a plugin.

Normal response codes: 200

5.1.2.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{plugin_name}	String	Name of the plugin.

This operation does not accept a request body.

5.1.2.2. Response

Example 5.2. 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."
  }
}
```

5.1.3. 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

5.1.3.1. Request

This table shows the URI parameters for the show plugin version details request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{plugin_name}	String	Name of the plugin.
{version}	String	Version of the plugin.

This operation does not accept a request body.

5.1.3.2. Response

Example 5.3. 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"
}
```

{}

5.2. Image registry

The image registry is a tool for managing images. Each plugin lists required tags for an image. The Data Processing service also requires a user name to log in to an instance's OS for remote operations execution.

The image registry enables you to add tags to and remove tags from images and define the OS user name.

Method	URI	Description
GET	/v1.1/{tenant_id}/images	Lists all images registered in the image registry.
GET	/v1.1/{tenant_id}/images/{image_id}	Shows details about an image.
POST	/v1.1/{tenant_id}/images/{image_id}	Registers an image in the image registry.
DELETE	/v1.1/{tenant_id}/images/{image_id}	Removes an image from the image 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.

5.2.1. List images

Method	URI	Description
GET	/v1.1/{tenant_id}/images	Lists all images registered in the image registry.

Normal response codes: 200

5.2.1.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

This operation does not accept a request body.

5.2.1.2. Response

Example 5.4. 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": 10000000000
    }
  ]
}
```

```
    "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
}
]
}
```

5.2.2. Show image details

Method	URI	Description
GET	/v1.1/{tenant_id}/images/{image_id}	Shows details about an image.

Normal response codes: 200

5.2.2.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{image_id}	UUID	The unique identifier of the image.

This operation does not accept a request body.

5.2.2.2. Response

Example 5.5. 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
  }
}
```

5.2.3. Register image

Method	URI	Description
POST	/v1.1/{tenant_id}/images/{image_id}	Registers an image in the image registry.

Normal response codes: 202

5.2.3.1. Request

This table shows the URI parameters for the register image request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{image_id}	UUID	The unique identifier of the image.

Example 5.6. Register image: JSON request

```
{
    "username": "ubuntu",
    "description": "Ubuntu image for Hadoop 2.7.1"
}
```

5.2.3.2. Response

Example 5.7. 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
    }
}
```

5.2.4. Unregister image

Method	URI	Description
DELETE	/v1.1/{tenant_id}/images/{image_id}	Removes an image from the image registry.

Normal response codes: 204

5.2.4.1. Request

This table shows the URI parameters for the unregister image request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{image_id}	UUID	The unique identifier of the image.

This operation does not accept a request body.

5.2.5. 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

5.2.5.1. Request

This table shows the URI parameters for the add tags to image request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{image_id}	UUID	The unique identifier of the image.

Example 5.8. Add tags to image: JSON request

```
{
  "tags": [
    "vanilla",
    "2.7.1",
    "some_other_tag"
  ]
}
```

5.2.5.2. Response

Example 5.9. 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
  }
}
```

```
    }  
}
```

5.2.6. 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

5.2.6.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{image_id}	UUID	The unique identifier of the image.

Example 5.10. Remove tags from image: JSON request

```
{
  "tags": [
    "some_other_tag"
  ]
}
```

5.2.6.2. Response

Example 5.11. 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
  }
}
```

5.3. 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.

5.3.1. List node group templates

Method	URI	Description
GET	/v1.1/{tenant_id}/node-group-templates	Lists available node group templates.

Normal response codes: 200

5.3.1.1. Request

This table shows the URI parameters for the list node group templates request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

This operation does not accept a request body.

5.3.1.2. Response

Example 5.12. 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,
      "is_standalone": 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
    }
]
```

5.3.2. Create node group template

Method	URI	Description
POST	/v1.1/{tenant_id}/node-group-templates	Creates a node group template.

Normal response codes: 202

5.3.2.1. Request

This table shows the URI parameters for the create node group template request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

Example 5.13. 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"
}
```

5.3.2.2. Response

Example 5.14. 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
    }
}
```

5.3.3. 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

5.3.3.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{node_group_template_id}	UUID	The unique identifier of the node group template.

This operation does not accept a request body.

5.3.3.2. Response

Example 5.15. 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
    }
}
```

5.3.4. 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

5.3.4.1. Request

This table shows the URI parameters for the delete node group template request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{node_group_template_id}	UUID	The unique identifier of the node group template.

This operation does not accept a request body.

5.3.5. 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

5.3.5.1. Request

This table shows the URI parameters for the update node group template request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{node_group_template_id}	UUID	The unique identifier of the node group template.

Example 5.16. 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"
}
```

5.3.5.2. Response

Example 5.17. 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"
    ],
    "node_configs": {}
  }
}
```

```
        "node_processes": [
            "datanode"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "volume_type": null
    }
}
```

5.4. 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 the cluster template.
PUT	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Updates the cluster template.
DELETE	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Deletes the cluster template.

5.4.1. List available cluster templates

Method	URI	Description
GET	/v1.1/{tenant_id}/cluster-templates	Lists available cluster templates.

Normal response codes: 200

5.4.1.1. Request

This table shows the URI parameters for the list available cluster templates request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

This operation does not accept a request body.

5.4.1.2. Response

Example 5.18. List available 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-
bf31-660cfaf5621bd",
    "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
}
]
```

5.4.2. Create cluster templates

Method	URI	Description
POST	/v1.1/{tenant_id}/cluster-templates	Creates a cluster template.

Normal response codes: 202

5.4.2.1. Request

This table shows the URI parameters for the create cluster templates request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

Example 5.19. 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"
}
```

5.4.2.2. Response

Example 5.20. 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,
        "name": "cluster-template"
      }
    ]
  }
}
```

```
"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
    }
}
```

5.4.3. Show cluster template

Method	URI	Description
GET	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Shows the cluster template.

Normal response codes: 200

5.4.3.1. Request

This table shows the URI parameters for the show cluster template request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{cluster_template_id}	UUID	The unique identifier of the cluster template.

This operation does not accept a request body.

5.4.3.2. Response

Example 5.21. Show cluster template: 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-
bf31-660cf5621bd",
    "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": {}
    },
    "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
    }
]
```

5.4.4. Update cluster templates

Method	URI	Description
PUT	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Updates the cluster template.

Normal response codes: 202

5.4.4.1. Request

This table shows the URI parameters for the update cluster templates request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{cluster_template_id}	UUID	The unique identifier of the cluster template.

Example 5.22. 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
        }
    }
}
```

5.4.4.2. Response

Example 5.23. 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
}
}
```

5.4.5. Delete cluster template

Method	URI	Description
DELETE	/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}	Deletes the cluster template.

Normal response codes: 204

5.4.5.1. Request

This table shows the URI parameters for the delete cluster template request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{cluster_template_id}	UUID	The unique identifier of the cluster template.

This operation does not accept a request body.

5.5. 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 cluster by ID.
DELETE	/v1.1/{tenant_id}/clusters/{cluster_id}	Deletes a cluster.
PUT	/v1.1/{tenant_id}/clusters/{cluster_id}	Scales a cluster.
PATCH	/v1.1/{tenant_id}/clusters/{cluster_id}	Updates a cluster.

5.5.1. List available clusters

Method	URI	Description
GET	/v1.1/{tenant_id}/clusters	Lists available clusters.

Normal response codes: 200

5.5.1.1. Request

This table shows the URI parameters for the list available clusters request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

This operation does not accept a request body.

5.5.1.2. Response

Example 5.24. 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
AAAAB3NzaC1yc2EAAAQABAAQCe9ARO
+t9CybtuCl+cusDTeQL7wos1+U2dKP1CUJvNUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rj0wAQANhL7KatGLyCPQg9fqOoaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZhTOvEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9PhoSqcrRbm72WzgLyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpduFLH4eN5MFh 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-9ala-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": "0cflee81-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",
    "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
    "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-aalcf9623e3e",
    "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"
    }
}
```

5.5.2. Create cluster

Method	URI	Description
POST	/v1.1/{tenant_id}/clusters	Creates a cluster.

Normal response codes: 202

5.5.2.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

Example 5.25. 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"
}
```

5.5.2.2. Response

Example 5.26. 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
AAAAB3NzaC1yc2EAAAQABAAQCe9ARO
+pt9CybtuCl+cusDTeQL7wos1+U2dKP1CUJvNUUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rj0wAQAnhL7KatGLyCPQg9fqOoaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZhTOvEog1LwZCzoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9PhoSqcrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpduFLH4eN5MFh 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"
}
}
```

5.5.3. Create multiple clusters

Method	URI	Description
POST	/v1.1/{tenant_id}/clusters/multiple	Creates multiple clusters.

Normal response codes: 202

5.5.3.1. Request

This table shows the URI parameters for the create multiple clusters request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

Example 5.27. 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"
}
```

5.5.3.2. Response

Example 5.28. Create multiple clusters: JSON response

```
{
    "clusters": [
        "a007a3e7-658f-4568-b0f2-fe2fd5efc554",
        "b012a6et-65hf-4566-b0f2-fe3fd7efc567"
    ]
}
```

5.5.4. Show details of a cluster

Method	URI	Description
GET	/v1.1/{tenant_id}/clusters/{cluster_id}	Shows cluster by ID.

Normal response codes: 200

5.5.4.1. Request

This table shows the URI parameters for the show details of a cluster request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{cluster_id}	UUID	The ID of the cluster

This operation does not accept a request body.

5.5.4.2. Response

Example 5.29. 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-bf31-660cfaf5621bd",
    "info": {},
    "user_keypair_id": "test",
    "management_public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQCe9ARO
+t9CybtuCl+cusDTeQL7wos1+U2dKPlCUJvNUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rj0wAQANhL7KatGLyCPQg9fqOoaiyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZhTOvEog1LwZCz0Tt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9PhoSqcrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFUvpduFLH4eN5MFh 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"
}
}
```

5.5.5. Delete a cluster

Method	URI	Description
DELETE	/v1.1/{tenant_id}/clusters/{cluster_id}	Deletes a cluster.

Normal response codes: 204

5.5.5.1. Request

This table shows the URI parameters for the delete a cluster request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{cluster_id}	UUID	The ID of the cluster

This operation does not accept a request body.

5.5.6. Scale cluster

Method	URI	Description
PUT	/v1.1/{tenant_id}/clusters/{cluster_id}	Scales a cluster.

Normal response codes: 202

5.5.6.1. Request

This table shows the URI parameters for the scale cluster request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{cluster_id}	UUID	The ID of the cluster

Example 5.30. 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"
    }
  ]
}
```

5.5.6.2. Response

Example 5.31. 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
AAAAB3NzaC1yc2EAAAQABAAQCe9ARO
+t9CybtuCl+cusDTeQL7wos1+U2dKP1CUJvNUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rj0wAQANhL7KatGLyCPQg9fqOoaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZkTOvEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9PhoSqcrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFUvpdutFLH4eN5MFh 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": "36flefde-90f9-41c1-b409-aalcf9623e3e"
},
{
"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-bf31-660cfaf5621bd",
"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_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-9ala-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": "0cflee81-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"  
    }  
}
```

5.5.7. Update cluster

Method	URI	Description
PATCH	/v1.1/{tenant_id}/clusters/{cluster_id}	Updates a cluster.

Normal response codes: 202

5.5.7.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{cluster_id}	UUID	The ID of the cluster

Example 5.32. Update cluster : JSON request

```
{
    "name": "public-vanilla-cluster",
    "is_public": true
}
```

5.5.7.2. Response

Example 5.33. 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
AAAAB3NzaC1yc2EAAAQABAAQCfe9ARO
+t9CybtuCl+cusDTeQL7wos1+U2dKPlCUJvNUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rj0wAQANhL7KatGLyCPQg9fqOoaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZhTOvEog1LwZCzoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9PhoSqcrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpduFLH4eN5MFDr 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"
}
}

```

5.6. 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.

5.6.1. Show progress

Method	URI	Description
GET	/v1.1/{tenant_id}/clusters/{cluster_id}	Shows provisioning progress of cluster.

Normal response codes: 200

5.6.1.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{cluster_id}	UUID	The ID of the cluster

5.6.1.2. Response

Example 5.34. 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"
                }
            ]
        }
    ]
}
```

```

        "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"
}
}

```

5.7. 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.
DELETE	/v1.1/{tenant_id}/data-sources/{data_source_id}	Deletes a data source.
PUT	/v1.1/{tenant_id}/data-sources/{data_source_id}	Updates a data source.

5.7.1. List data sources

Method	URI	Description
GET	/v1.1/{tenant_id}/data-sources	Lists all data sources.

Normal response codes: 200

5.7.1.1. Request

This table shows the URI parameters for the list data sources request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

This operation does not accept a request body.

5.7.1.2. Response

Example 5.35. 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": "d7ffffe9c-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"
    }
  ]
}
```

5.7.2. Create data source

Method	URI	Description
POST	/v1.1/{tenant_id}/data-sources	Creates a data source.

Normal response codes: 202

5.7.2.1. Request

This table shows the URI parameters for the create data source request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

Example 5.36. Create data source: JSON request

```
{
  "description": "This is input",
  "url": "swift://container/text",
  "credentials": {
    "password": "swordfish",
    "user": "dev"
  },
  "type": "swift",
  "name": "swift_input"
}
```

Example 5.37. Create data source: JSON request

```
{
  "description": "This is hdfs input",
  "url": "hdfs://test-master-node:8020/user/hadoop/input",
  "type": "hdfs",
  "name": "hdfs_input"
}
```

5.7.2.2. Response

Example 5.38. 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 5.39. 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": "d7ffffe9c-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"  
    }  
}
```

5.7.3. 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

5.7.3.1. Request

This table shows the URI parameters for the show data source details request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{data_source_id}	UUID	The unique identifier of the data source.

This operation does not accept a request body.

5.7.3.2. Response

Example 5.40. 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"
  }
}
```

5.7.4. Delete data source

Method	URI	Description
DELETE	/v1.1/{tenant_id}/data-sources/{data_source_id}	Deletes a data source.

Normal response codes: 204

5.7.4.1. Request

This table shows the URI parameters for the delete data source request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{data_source_id}	UUID	The unique identifier of the data source.

This operation does not accept a request body.

5.7.5. Update data source

Method	URI	Description
PUT	/v1.1/{tenant_id}/data-sources/{data_source_id}	Updates a data source.

Normal response codes: 202

5.7.5.1. Request

This table shows the URI parameters for the update data source request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{data_source_id}	UUID	The unique identifier of the data source.

Example 5.41. Update data source : JSON request

```
{
    "description": "This is public input",
    "is_protected": true
}
```

5.7.5.2. Response

Example 5.42. 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"
    }
}
```

5.8. 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.

Method	URI	Description
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	Gets data for a job binary internal.

5.8.1. 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

5.8.1.1. Request

This table shows the URI parameters for the list job binary internals request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

This operation does not accept a request body.

5.8.1.2. Response

Example 5.43. 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
    }
  ]
}
```

5.8.2. 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

5.8.2.1. Request

This table shows the URI parameters for the create job binary internal request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{name}	String	The name of the job binary internal.

This operation does not accept a request body.

5.8.2.2. Response

Example 5.44. 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
  }
}
```

5.8.3. 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

5.8.3.1. Request

This table shows the URI parameters for the show job binary internal details request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_binary_internals_id}	UUID	The ID of the job binary internal.

This operation does not accept a request body.

5.8.3.2. Response

Example 5.45. 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
    }
}
```

5.8.4. 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

5.8.4.1. Request

This table shows the URI parameters for the delete job binary internal request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_binary_internals_id}	UUID	The ID of the job binary internal.

This operation does not accept a request body.

5.8.5. 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

5.8.5.1. Request

This table shows the URI parameters for the update job binary internal request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_binary_internals_id}	UUID	The ID of the job binary internal.

Example 5.46. Update job binary internal : JSON request

```
{
  "name": "public-jbi",
  "is_public": true
}
```

5.8.5.2. Response

Example 5.47. 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
  }
}
```

5.8.6. Get job binary internal data

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}/data	Gets 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 2015 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

5.8.6.1. Request

This table shows the URI parameters for the get job binary internal data request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_binary_internals_id}	UUID	The ID of the job binary internal.

This operation does not accept a request body.

5.8.6.2. Response

This operation does not return a response body.

5.9. 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.
POST	/v1.1/{tenant_id}/job-binaries	Creates a job binary.
GET	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}	Shows details for a job binary.
DELETE	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}	Deletes a job binary.
PUT	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}	Updates a job binary.

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}/data	Gets data for a job binary.

5.9.1. List job binaries

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binaries	Lists the available job binaries.

Normal response codes: 200

5.9.1.1. Request

This table shows the URI parameters for the list job binaries request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

This operation does not accept a request body.

5.9.1.2. Response

Example 5.48. 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
    }
  ]
}
```

```
    ]  
}
```

5.9.2. Create job binary

Method	URI	Description
POST	/v1.1/{tenant_id}/job-binaries	Creates a job binary.

Normal response codes: 202

5.9.2.1. Request

This table shows the URI parameters for the create job binary request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

Example 5.49. 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"
    }
}
```

5.9.2.2. Response

Example 5.50. 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
    }
}
```

5.9.3. Show job binary details

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}	Shows details for a job binary.

Normal response codes: 200

5.9.3.1. Request

This table shows the URI parameters for the show job binary details request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_binaries_id}	UUID	The ID of the job binary.

This operation does not accept a request body.

5.9.3.2. Response

Example 5.51. 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
  }
}
```

5.9.4. Delete job binary

Method	URI	Description
DELETE	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}	Deletes a job binary.

Normal response codes: 204

5.9.4.1. Request

This table shows the URI parameters for the delete job binary request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_binaries_id}	UUID	The ID of the job binary.

This operation does not accept a request body.

5.9.5. Update job binary

Method	URI	Description
PUT	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}	Updates a job binary.

Normal response codes: 202

5.9.5.1. Request

This table shows the URI parameters for the update job binary request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_binaries_id}	UUID	The ID of the job binary.

Example 5.52. Update job binary : JSON request

```
{
  "url": "swift://container/new-jar-example.jar",
  "name": "new-jar-example.jar",
  "description": "This is a new job binary"
}
```

5.9.5.2. Response

Example 5.53. 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
  }
}
```

5.9.6. Get job binary data

Method	URI	Description
GET	/v1.1/{tenant_id}/job-binaries/{job_binaries_id}/data	Gets 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 2015 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

5.9.6.1. Request

This table shows the URI parameters for the get job binary data request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_binaries_id}	UUID	The ID of the job binary.

This operation does not accept a request body.

5.9.6.2. Response

This operation does not return a response body.

5.10. 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.

Method	URI	Description
POST	/v1.1/{tenant_id}/jobs/{job_id}/execute	Runs a job.

5.10.1. List jobs

Method	URI	Description
GET	/v1.1/{tenant_id}/jobs	Lists all jobs.

Normal response codes: 200

5.10.1.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

This operation does not accept a request body.

5.10.1.2. Response

Example 5.54. 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
]
}
```

5.10.2. Create job

Method	URI	Description
POST	/v1.1/{tenant_id}/jobs	Creates a job object.

Normal response codes: 202

5.10.2.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

Example 5.55. 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"
}
```

5.10.2.2. Response

Example 5.56. 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"  
}  
]  
}
```

5.10.3. Show job details

Method	URI	Description
GET	/v1.1/{tenant_id}/jobs/{job_id}	Shows details for a job.

Normal response codes: 200

5.10.3.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_id}	UUID	The unique identifier of the job.

This operation does not accept a request body.

5.10.3.2. Response

Example 5.57. 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
  }
}
```

5.10.4. Remove job

Method	URI	Description
DELETE	/v1.1/{tenant_id}/jobs/{job_id}	Removes a job.

Normal response codes: 204

5.10.4.1. Request

This table shows the URI parameters for the remove job request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_id}	UUID	The unique identifier of the job.

This operation does not accept a request body.

5.10.5. Update job object

Method	URI	Description
PATCH	/v1.1/{tenant_id}/jobs/{job_id}	Updates a job object.

Normal response codes: 202

5.10.5.1. Request

This table shows the URI parameters for the update job object request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_id}	UUID	The unique identifier of the job.

Example 5.58. Update job object: JSON request

```
{
  "description": "This is public pig job example",
  "name": "public-pig-job-example"
}
```

5.10.5.2. Response

Example 5.59. 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
  }
}
```

5.10.6. Run job

Method	URI	Description
POST	/v1.1/{tenant_id}/jobs/{job_id}/execute	Runs a job.

Normal response codes: 202

5.10.6.1. Request

This table shows the URI parameters for the run job request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_id}	UUID	The unique identifier of the job.

Example 5.60. 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"
    }
  }
}
```

5.10.6.2. Response

Example 5.61. 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.map.tasks": "1",
        "mapred.reduce.tasks": "1"
      }
    }
  }
}
```

```
        "mapred.reduce.tasks": "1",
        "mapred.map.tasks": "1"
    },
    "args": [
        "arg1",
        "arg2"
    ],
    "params": {
        "param2": "value2",
        "param1": "value1"
    }
},
"info": {
    "status": "PENDING"
}
}
```

5.11. 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.

5.11.1. List job executions

Method	URI	Description
GET	/v1.1/{tenant_id}/job-executions	Lists available job executions.

Normal response codes: 200

5.11.1.1. Request

This table shows the URI parameters for the list job executions request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.

This operation does not accept a request body.

5.11.1.2. Response

Example 5.62. 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-a473-41baeeefef997",
            "oozie_job_id": "0000001-150915094349962-oozie-hado-W",
            "status": "PENDING"
        }
    ]
}
```

```
"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"
}
}
]
```

5.11.2. 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

5.11.2.1. Request

This table shows the URI parameters for the show job execution details request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_execution_id}	UUID	The unique identifier of the job execution.

This operation does not accept a request body.

5.11.2.2. Response

Example 5.63. 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": {
            "status": "PENDING",
            "progress": 0
        }
    }
}
```

```
"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]"
    }
]
```

```
        "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"
}
}
}
```

5.11.3. Delete job execution

Method	URI	Description
DELETE	/v1.1/{tenant_id}/job-executions/{job_execution_id}	Deletes a job execution.

Normal response codes: 204

5.11.3.1. Request

This table shows the URI parameters for the delete job execution request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_execution_id}	UUID	The unique identifier of the job execution.

This operation does not accept a request body.

5.11.4. Update job execution

Method	URI	Description
PATCH	/v1.1/{tenant_id}/job-executions/{job_execution_id}	Updates a job execution.

Normal response codes: 202

5.11.4.1. Request

This table shows the URI parameters for the update job execution request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_execution_id}	UUID	The unique identifier of the job execution.

Example 5.64. Update job execution: JSON request

```
{
    "is_public": true
}
```

5.11.4.2. Response

Example 5.65. 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": "808d5032ea0446889097723bf8e919d",
    }
}
```

```
"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"
}
}
}
```

5.11.5. 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

5.11.5.1. Request

This table shows the URI parameters for the refresh job execution status request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_execution_id}	UUID	The unique identifier of the job execution.

This operation does not accept a request body.

5.11.5.2. Response

Example 5.66. 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\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"
}
}
}
```

5.11.6. 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

5.11.6.1. Request

This table shows the URI parameters for the cancel job execution request:

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{job_execution_id}	UUID	The unique identifier of the job execution.

This operation does not accept a request body.

5.11.6.2. Response

Example 5.67. 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": {
            "status": "CANCELED"
        }
    }
}
```

```
"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]"
    }
]
```

```

        "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"
}
}
}
}

```

5.12. 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 job types.

5.12.1. List job types

Method	URI	Description
GET	/v1.1/{tenant_id}/job-types	Lists job types.

You can filter the response through filter parameters.

Normal response codes: 200

5.12.1.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The unique identifier of the tenant or account.
{plugin}	String	Filter results by plugin name.
{version}	String	Filter results by plugin version.
{type}	String	Filter results by job type.
{hints}	Bool	Include configuration hints in results.

This operation does not accept a request body.

5.12.1.2. Response

Example 5.68. 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"
}
]
```

6. Identity API v3 (CURRENT)

Gets an authentication token that permits access to the OpenStack services REST API.

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*.

Method	URI	Description
API versions		
GET	/	Lists information about all Identity API versions.
GET	/v3	Shows details for the Identity API v3.
Tokens		
POST	/v3/auth/tokens	Authenticates and generates a token.
GET	/v3/auth/tokens	Validates a token.
HEAD	/v3/auth/tokens	Validates a token.
DELETE	/v3/auth/tokens	Revokes a token.
Service catalog		
POST	/v3/services	Creates a service.
GET	/v3/services{?type}	Lists services.
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.
Endpoints		
POST	/v3/endpoints	Creates an endpoint.
GET	/v3/endpoints{?interface, service_id}	Lists available endpoints.
PATCH	/v3/endpoints/{endpoint_id}	Updates an endpoint.
DELETE	/v3/endpoints/{endpoint_id}	Deletes an endpoint.
Domains		
POST	/v3/domains	Creates a domain.
GET	/v3/domains{?name, enabled}	Lists domains.
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.
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 domain user.
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 domain user.

Method	URI	Description
GET	/v3/domains/{domain_id}/groups/{group_id}/roles	Lists roles for a domain group.
PUT	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Grants a role to a domain group.
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.
Projects		
POST	/v3/projects	Creates a project.
GET	/v3/projects{?domain_id,parent_id,name(enabled)}	Lists projects.
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.
GET	/v3/projects/{project_id}/users/{user_id}/roles	Lists roles for a user in a project.
PUT	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Grants a role to a user in a project.
HEAD	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Validates that a user has a role in a project.
DELETE	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Revokes a role from a project user.
GET	/v3/projects/{project_id}/groups/{group_id}/roles	Lists roles for a project group.
PUT	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Grants a role to a project group.
HEAD	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Validates that a project group has a role.
DELETE	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Revokes a role from a project group.
Users		
POST	/v3/users	Creates a user.
GET	/v3/users{?domain_id,name(enabled)}	Lists users.
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.
GET	/v3/users/{user_id}/groups	Lists groups for a user.
GET	/v3/users/{user_id}/projects	List projects for a user.
Groups		
POST	/v3/groups	Creates a group.
GET	/v3/groups{?domain_id}	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{?name, domain_id,description, name(enabled)}	Lists users in a group.
PUT	/v3/groups/{group_id}/users/{user_id}	Assigns a user to a group.

Method	URI	Description
DELETE	/v3/groups/{group_id}/users/{user_id}	Removes a user from a group.
HEAD	/v3/groups/{group_id}/users/{user_id}	Validates that a user is in a group.
Credentials		
POST	/v3/credentials	Creates a credential.
GET	/v3/credentials{?user_id}	Lists 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.
Roles		
POST	/v3/roles	Creates a role.
GET	/v3/roles{?name}	Lists roles.
GET	/v3/role_assignments{?group.id, role.id, scope.domain.id, scope.project.id, user.id, effective}	Lists role assignments.
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.
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.
Regions		
GET	/v3/regions{?parent_region_id}	Lists regions.
POST	/v3/regions	Creates a region.
PUT	/v3/regions/{user_defined_region_id}	Creates a region with a user-defined region ID.
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.

6.1. API versions

Method	URI	Description
GET	/	Lists information about all Identity API versions.
GET	/v3	Shows details for the Identity API v3.

6.1.1. List versions

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

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.1.1.1. Request

This operation does not accept a request body.

6.1.1.2. Response

Example 6.1. 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+json"
            }
          ]
        }
      ]
    }
  ]
}
```

```
        "type": "application/vnd.openstack.identity-v2.0+json"
    }
],
"status": "stable",
"updated": "2014-04-17T00:00:00Z"
}
]
}
```

6.1.2. Show API version details

Method	URI	Description
GET	/v3	Shows details for the Identity API v3.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404),

6.1.2.1. Request

This operation does not accept a request body.

6.1.2.2. Response

Example 6.2. Show API version details: JSON response

```
{
  "version": {
    "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"
  }
}
```

6.2. Tokens

Manages tokens.

Method	URI	Description
POST	/v3/auth/tokens	Authenticates and generates a token.
GET	/v3/auth/tokens	Validates a token.
HEAD	/v3/auth/tokens	Validates a token.
DELETE	/v3/auth/tokens	Revokes a token.

6.2.1. Authenticate

Method	URI	Description
POST	/v3/auth/tokens	Authenticates and generates a token.

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.

A REST interface provides client authentication by using the **POST** method with auth/tokens the path. The body of the request must include a payload of credentials including the authentication method and, optionally, the authorization scope. The scope includes either a project or domain. If you include both project and domain, this call returns the HTTP Bad Request (400) status code because a token cannot be simultaneously scoped as both a project and domain.

Important



If you do not include the optional scope and the authenticating user has a defined default project (the default_project_id attribute for the user), that default project is treated as the preferred authorization scope.

If no default project is defined, the token is issued without an explicit scope of authorization.

Provide one of the following sets of credentials to authenticate: User ID and password, user name and password scoped by domain ID or name, user ID and password scoped by project ID or name with or without domain scope, or token.

The following examples demonstrate authentication requests with different types of credentials.

Note



If scope is included, project id uniquely identifies the project. However, project name uniquely identifies the project only when used in conjunction with a domain ID or a domain name.

If the authentication token has expired, this call returns the HTTP 401 status code.

If the subject 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.

As the following example responses show, the response to an authentication request returns the token ID in the X-Subject-Token header instead of in the token data.

If the call has no explicit authorization scope, the response does not contain the catalog, project, domain, or roles fields. However, the response still uniquely identifies the user.

A token scoped to a project also has both a service catalog and the user's roles applicable to the project.

A token scoped to a domain also has both a service catalog and the user's roles applicable to the project.

Optionally, the Identity API implementation might return an authentication attribute to indicate the supported authentication methods.

For authentication processes that require multiple round trips, The Identity API implementation might return an HTTP Unauthorized (401) status code with additional information for the next authentication step.

The following examples illustrate several possible HTTP Unauthorized (401) authentication errors. Other errors, like HTTP Forbidden (403), are also possible.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.2.1.1. Request

Example 6.3. Authenticate: JSON request

```
{  
  "auth": {  
    "identity": {  
      "methods": [  
        "password"  
      ],  
      "password": {  
        "user": {  
          "id": "0ca8f6",  
          "password": "secretsecret"  
        }  
      }  
    }  
  }  
}
```

Example 6.4. Authenticate: JSON request

```
{  
  "auth": {  
    "identity": {  
      "methods": [  
        "password"  
      ],  
      "password": {  
        "user": {  
          "domain": {  
            "id": "1789d1"  
          },  
          "name": "Joe",  
          "password": "secretsecret"  
        }  
      }  
    }  
  }  
}
```

```
        }
    }
}
```

Example 6.5. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "domain": {
            "name": "example.com"
          },
          "name": "Joe",
          "password": "secretsecret"
        }
      }
    }
  }
}
```

Example 6.6. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "token"
      ],
      "token": {
        "id": "'$OS_TOKEN'"
      }
    }
  }
}
```

Example 6.7. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "id": "0ca8f6",
          "password": "secretsecret"
        }
      }
    },
    "scope": {
      "project": {
        "id": "263fd9"
      }
    }
  }
}
```

```
        }
    }
}
```

Example 6.8. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "id": "0ca8f6",
          "password": "secretsecret"
        }
      }
    },
    "scope": {
      "domain": {
        "id": "263fd9"
      }
    }
  }
}
```

Example 6.9. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "id": "0ca8f6",
          "password": "secretsecret"
        }
      }
    },
    "scope": {
      "project": {
        "domain": {
          "id": "1789d1"
        },
        "name": "project-x"
      }
    }
  }
}
```

Example 6.10. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [

```

```
        "password"
    ],
    "password": {
        "user": {
            "id": "0ca8f6",
            "password": "secretsecret"
        }
    }
},
"scope": {
    "project": {
        "domain": {
            "name": "example.com"
        },
        "name": "project-x"
    }
}
}
```

6.2.1.2. Response

Example 6.11. Authenticate: JSON response

```
{
    "token": {
        "methods": [
            "password"
        ],
        "roles": [
            {
                "id": "9fe2ff9ee4384b1894a90878d3e92bab",
                "name": "_member_"
            },
            {
                "id": "c703057be878458588961ce9a0ce686b",
                "name": "admin"
            }
        ],
        "expires_at": "2014-06-10T2:55:16.806001Z",
        "project": {
            "domain": {
                "id": "default",
                "name": "Default"
            },
            "id": "8538a3f13f9541b28c2620eb19065e45",
            "name": "admin"
        },
        "catalog": [
            {
                "endpoints": [
                    {
                        "url": "http://localhost:3537/v2.0",
                        "region": "RegionOne",
                        "interface": "admin",
                        "id": "29beb2f1567642eb810b042b6719ea88"
                    },
                    {
                        "url": "http://localhost:5000/v2.0",
                        "region": "RegionOne",
                        "interface": "admin"
                    }
                ]
            }
        ]
    }
}
```

```
        "interface": "internal",
        "id": "8707e3735d4415c97ae231b4841eb1c"
    },
    {
        "url": "http://localhost:5000/v2.0",
        "region": "RegionOne",
        "interface": "public",
        "id": "ef303187fc8d41668f25199c298396a5"
    }
],
"type": "identity",
"id": "bd73972c0e14fb69bae8ff76e112a90",
"name": "keystone"
}
],
"extras": {},
"user": {
    "domain": {
        "id": "default",
        "name": "Default"
    },
    "id": "3ec3164f750146be97f21559ee4d9c51",
    "name": "admin"
},
"audit_ids": [
    "yRt0UrxJSs6-WYJgwEMMmg"
],
"issued_at": "201406-10T20:55:16.806027Z"
}
}
```

Example 6.12. Authenticate: JSON response

```
{
    "token": {
        "audit_ids": [
            "ECwrVNWbSCqmEgPnu0YCRw"
        ],
        "methods": [
            "password"
        ],
        "roles": [
            {
                "id": "c703057be878458588961ce9a0ce686b",
                "name": "admin"
            }
        ],
        "expires_at": "2014-06-10T21:40:14.360795Z",
        "project": {
            "domain": {
                "id": "default",
                "name": "Default"
            },
            "id": "3d4c2c82bd5948f0bcab0cf3a7c9b48c",
            "name": "demo"
        },
        "catalog": [
            {
                "endpoints": [
                    {
                        "interface": "internal",
                        "id": "8707e3735d4415c97ae231b4841eb1c"
                    },
                    {
                        "url": "http://localhost:5000/v2.0",
                        "region": "RegionOne",
                        "interface": "public",
                        "id": "ef303187fc8d41668f25199c298396a5"
                    }
                ],
                "type": "identity",
                "id": "bd73972c0e14fb69bae8ff76e112a90",
                "name": "keystone"
            }
        ],
        "extras": {},
        "user": {
            "domain": {
                "id": "default",
                "name": "Default"
            },
            "id": "3ec3164f750146be97f21559ee4d9c51",
            "name": "admin"
        },
        "audit_ids": [
            "yRt0UrxJSs6-WYJgwEMMmg"
        ],
        "issued_at": "201406-10T20:55:16.806027Z"
    }
}
```

```
        "url": "http://localhost:35357/v2.0",
        "region": "RegionOne",
        "interface": "admin",
        "id": "29beb2f1567642eb810b042b6719ea88"
    },
    {
        "url": "http://localhost:5000/v2.0",
        "region": "RegionOne",
        "interface": "internal",
        "id": "87057e3735d4415c97ae231b4841eb1c"
    },
    {
        "url": "http://localhost:5000/v2.0",
        "region": "RegionOne",
        "interface": "public",
        "id": "ef303187fc8d41668f25199c298396a5"
    }
],
"type": "identity",
"id": "bd7397d2c0e14fb69bae8ff76e112a90",
"name": "keystone"
}
],
"extras": {},
"user": {
    "domain": {
        "id": "default",
        "name": "Default"
    },
    "id": "3ec3164f750146be97f21559ee4d9c51",
    "name": "admin"
},
"issued_at": "2014-06-10T20:40:14.360822Z"
}
}
```

Example 6.13. Authenticate: JSON response

```
{
    "token": {
        "domain": {
            "id": "default",
            "name": "Default"
        },
        "methods": [
            "password"
        ],
        "roles": [
            {
                "id": "c703057be878458588961ce9a0ce686b",
                "name": "admin"
            }
        ],
        "expires_at": "2014-06-10T21:52:58.852167Z",
        "catalog": [
            {
                "endpoints": [
                    {
                        "url": "http://localhost:35357/v2.0",
                        "region": "RegionOne",

```

```

        "interface": "admin",
        "id": "29beb2f1567642eb810b042b6719ea88"
    },
    {
        "url": "http://localhost:5000/v2.0",
        "region": "RegionOne",
        "interface": "internal",
        "id": "87057e3735d4415c97ae231b4841eb1c"
    },
    {
        "url": "http://localhost:5000/v2.0",
        "region": "RegionOne",
        "interface": "public",
        "id": "ef303187fc8d41668f25199c298396a5"
    }
],
{
    "type": "identity",
    "id": "bd7397d2c0e14fb69bae8ff76e112a90",
    "name": "keystone"
}
],
"extras": {},
"user": {
    "domain": {
        "id": "default",
        "name": "Default"
    },
    "id": "3ec3164f750146be97f21559ee4d9c51",
    "name": "admin"
},
"audit_ids": [
    "Xpa6Uyn-T9S6mTREudUH3w"
],
"issued_at": "2014-06-10T20:52:58.852194Z"
}
}
}

```

Example 6.14. Authenticate: JSON response

```
{
    "token": {
        "methods": [
            "token",
            "password"
        ],
        "expires_at": "2015-05-28T07:43:44.808209Z",
        "extras": {},
        "user": {
            "domain": {
                "id": "default",
                "name": "Default"
            },
            "id": "753867c25c3340ffad1abc22d488c31a",
            "name": "admin"
        },
        "audit_ids": [
            "ZE0OPSuzTmCXHo0eIOYltw",
            "xxIQCkHOQOywL0oY6CTppQ"
        ],
        "issued_at": "2015-05-28T07:19:23.763532Z"
    }
}
```

```
    }  
}
```

6.2.2. Validate token

Method	URI	Description
GET	/v3/auth/tokens	Validates a token.

Pass your own token in the X-Auth-Token header. Pass the token to be validated in the X-Subject-Token header. The Identity API returns the same response as when the subject token was issued by POST /auth/tokens.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.2.2.1. Request

This table shows the header parameters for the validate token request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.
X-Subject-Token	String <i>(Required)</i>	The token ID.

Example 6.15. Validate token: JSON request

```
Headers:
X-Auth-Token: 1dd7e3
X-Subject-Token: c67580
```

This operation does not accept a request body.

6.2.2.2. Response

Example 6.16. Validate token: JSON response

```
{
  "token": {
    "methods": [
      "password"
    ],
    "roles": [
      {
        "id": "9fe2ff9ee4384b1894a90878d3e92bab",
        "name": "_member_"
      },
      {
        "id": "c703057be878458588961ce9a0ce686b",
        "name": "admin"
      }
    ]
  }
}
```

```
"expires_at": "2014-06-10T2:55:16.806001Z",
"project": {
    "domain": {
        "id": "default",
        "name": "Default"
    },
    "id": "8538a3f13f9541b28c2620eb19065e45",
    "name": "admin"
},
"catalog": [
    {
        "endpoints": [
            {
                "url": "http://localhost:3537/v2.0",
                "region": "RegionOne",
                "interface": "admin",
                "id": "29beb2f1567642eb810b042b6719ea88"
            },
            {
                "url": "http://localhost:5000/v2.0",
                "region": "RegionOne",
                "interface": "internal",
                "id": "8707e3735d4415c97ae231b4841eb1c"
            },
            {
                "url": "http://localhost:5000/v2.0",
                "region": "RegionOne",
                "interface": "public",
                "id": "ef303187fc8d41668f25199c298396a5"
            }
        ],
        "type": "identity",
        "id": "bd73972c0e14fb69bae8ff76e112a90",
        "name": "keystone"
    }
],
"extras": {},
"user": {
    "domain": {
        "id": "default",
        "name": "Default"
    },
    "id": "3ec3164f750146be97f21559ee4d9c51",
    "name": "admin"
},
"audit_ids": [
    "yRt0UrxJSs6-WYJgwEMMmg"
],
"issued_at": "201406-10T20:55:16.806027Z"
}
}
```

6.2.3. 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 status code indicates that the X-Subject-Token is valid.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.2.3.1. Request

This table shows the header parameters for the check token request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.
X-Subject-Token	String <i>(Required)</i>	The token ID.

Example 6.17. Check token: JSON request

```
Headers:
X-Auth-Token: 1dd7e3
X-Subject-Token: c67580
```

This operation does not accept a request body.

6.2.4. Revoke token

Method	URI	Description
DELETE	/v3/auth/tokens	Revokes a token.

This call is similar to `HEAD /auth/tokens` 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.2.4.1. Request

This table shows the header parameters for the revoke token request:

Name	Type	Description
<code>X-Auth-Token</code>	String <i>(Required)</i>	A valid authentication token for an administrative user.
<code>X-Subject-Token</code>	String <i>(Required)</i>	The token ID.

Example 6.18. Revoke token: JSON request

```
Headers:
X-Auth-Token: 1dd7e3
X-Subject-Token: c67580
```

This operation does not accept a request body.

6.3. Service catalog

Manages the catalog of services.

Method	URI	Description
POST	/v3/services	Creates a service.
GET	/v3/services{?type}	Lists services.
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.

6.3.1. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.3.1.1. Request

This table shows the header parameters for the create service request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 6.19. Create service: JSON request

```
{
  "service": {
    "type": "volume"
  }
}
```

6.3.1.2. Response

Example 6.20. Create service: JSON response

```
{
  "service": {
    "enabled": true,
    "id": "686766",
    "links": {
      "self": "http://identity:5000/v3/services/686766"
    },
    "type": "volume"
  }
}
```

6.3.2. List services

Method	URI	Description
GET	/v3/services{?type}	Lists services.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.3.2.1. Request

This table shows the header parameters for the list services request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
type	String <i>(Optional)</i>	Filters by service type. Service types include compute, ec2, image, and identity.

This operation does not accept a request body.

6.3.2.2. Response

Example 6.21. List services: JSON response

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://identity:5000/v3/services"
  },
  "services": [
    {
      "description": "Keystone Identity Service",
      "enabled": true,
      "id": "686766",
      "links": {
        "self": "http://identity:5000/v3/services/686766"
      },
      "name": "keystone",
      "type": "identity"
    },
    {
      "enabled": true,
      "id": "936521",
      "links": {
        "self": "http://identity:5000/v3/services/936521"
      }
    }
  ]
}
```

```
        } ,  
        "type": "volume"  
    ]  
}
```

6.3.3. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.3.3.1. Request

This table shows the header parameters for the show service details request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

6.3.3.2. Response

Example 6.22. 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"
  }
}
```

6.3.4. Update service

Method	URI	Description
PATCH	/v3/services/{service_id}	Updates a service.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.3.4.1. Request

This table shows the header parameters for the update service request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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 6.23. Update service: JSON request

```
{
    "type": "volume"
}
```

6.3.4.2. Response

Example 6.24. Update service: JSON response

```
{
    "service": {
        "id": "686766",
        "type": "volume"
    }
}
```

6.3.5. Delete service

Method	URI	Description
DELETE	/v3/services/{service_id}	Deletes a service.



Warning

If you try to delete a service when endpoints exist, 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.3.5.1. Request

This table shows the header parameters for the delete service request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

6.4. Endpoints

Manages endpoints.

Method	URI	Description
POST	/v3/endpoints	Creates an endpoint.
GET	/v3/endpoints{?interface, service_id}	Lists available endpoints.
PATCH	/v3/endpoints/{endpoint_id}	Updates an endpoint.
DELETE	/v3/endpoints/{endpoint_id}	Deletes an endpoint.

6.4.1. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.4.1.1. Request

This table shows the header parameters for the create endpoint request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 6.25. Create endpoint: JSON request

```
{
  "endpoint": {
    "interface": "public",
    "name": "name",
    "region_id": "north",
    "url": "http://identity:35357/v3/endpoints/828384",
    "service_id": "686766"
  }
}
```

6.4.1.2. Response

Example 6.26. Create 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"
  }
}
```

6.4.2. List endpoints

Method	URI	Description
GET	/v3/endpoints{?interface, service_id}	Lists available endpoints.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.4.2.1. Request

This table shows the header parameters for the list endpoints request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
interface	String <i>(Optional)</i>	Filters by interface.
service_id	String <i>(Optional)</i>	Filters by service_id.

This operation does not accept a request body.

6.4.2.2. Response

Example 6.27. List endpoints: JSON response

```
{
  "endpoints": [
    {
      "id": "828384",
      "interface": "public",
      "links": {
        "self": "http://identity:35357/v3/endpoints/828384"
      },
      "name": "the public volume endpoint",
      "service_id": "686766"
    },
    {
      "id": "642136",
      "interface": "internal",
      "links": {
        "self": "http://identity:35357/v3/endpoints/642136"
      },
      "name": "the internal volume endpoint",
    }
  ]
}
```

```
        "service_id": "686766"
    }
}
```

6.4.3. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.4.3.1. Request

This table shows the header parameters for the update endpoint request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{endpoint_id}	Uuid	The endpoint ID.

Example 6.28. Update endpoint: JSON request

```
{
  "endpoint": {
    "interface": "public",
    "name": "Name",
    "region_id": "north",
    "url": "http://identity:35357/v3/endpoints/828384",
    "service_id": "345678"
  }
}
```

6.4.3.2. Response

Example 6.29. 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"
  }
}
```


6.4.4. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.4.4.1. Request

This table shows the header parameters for the delete endpoint request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.5. Domains

Manages domains.

Domains represent collections of users, groups, and projects. Each is owned by exactly one domain. Users, however, can be associated with multiple projects by granting roles to the user on a project, including projects owned by other domains.

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 the following 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
POST	/v3/domains	Creates a domain.

Method	URI	Description
GET	/v3/domains{?name,enabled}	Lists domains.
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.
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 domain user.
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 domain user.
GET	/v3/domains/{domain_id}/groups/{group_id}/roles	Lists roles for a domain group.
PUT	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Grants a role to a domain group.
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.

6.5.1. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.1.1. Request

This table shows the header parameters for the create domain request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 6.30. Create domain: JSON request

```
{
  "domain": {
    "description": "Domain description",
    "enabled": true,
    "name": "myDomain"
  }
}
```

6.5.1.2. Response

Example 6.31. Create domain: JSON response

```
{
  "domain": {
    "description": "Domain description",
    "enabled": true,
    "id": "161718",
    "links": {
      "self": "http://identity:35357/v3/domains/161718"
    },
    "name": "myDomain"
  }
}
```

6.5.2. List domains

Method	URI	Description
GET	/v3/domains{?name,enabled}	Lists domains.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.2.1. Request

This table shows the header parameters for the list domains request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
name	String <i>(Optional)</i>	Filters on a name.
enabled	String <i>(Optional)</i>	Filters on enabled or disabled roles. Values are true or false.

This operation does not accept a request body.

6.5.2.2. Response

Example 6.32. List domains: JSON response

```
{
  "domains": [
    {
      "description": "Domain description",
      "enabled": true,
      "id": "161718",
      "links": {
        "self": "http://identity:35357/v3/domains/161718"
      },
      "name": "myDomain"
    },
    {
      "description": "Another domain",
      "enabled": true,
      "id": "864369",
      "links": {
        "self": "http://identity:35357/v3/domains/864369"
      },
      "name": "anotherDomain"
    }
  ]
}
```

```
        }  
    ]  
}
```

6.5.3. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.3.1. Request

This table shows the header parameters for the show domain details request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.5.3.2. Response

Example 6.33. Show domain details: JSON response

```
{
  "domain": {
    "description": "Domain description",
    "enabled": true,
    "id": "161718",
    "links": {
      "self": "http://identity:35357/v3/domains/161718"
    },
    "name": "myDomain"
  }
}
```

6.5.4. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.4.1. Request

This table shows the header parameters for the update domain request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{domain_id}	Uuid	The domain ID.

Example 6.34. Update domain: JSON request

```
{
  "domain": {
    "description": "my updated domain",
    "enabled": true,
    "name": "myUpdatedDomain"
  }
}
```

6.5.4.2. Response

Example 6.35. Update domain: JSON response

```
{
  "domain": {
    "description": "my updated domain",
    "enabled": true,
    "id": "161718",
    "links": {
      "self": "http://identity:35357/v3/domains/161718"
    },
    "name": "myUpdatedDomain"
  }
}
```

6.5.5. 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. If you try to delete an enabled domain, this call returns an HTTP Forbidden (403) status code.



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.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.5.1. Request

This table shows the header parameters for the delete domain request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.5.6. List roles for domain user

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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.6.1. Request

This table shows the header parameters for the list roles for domain user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the list roles for domain user request:

Name	Type	Description
{domain_id}	Uuid	The domain ID.
{user_id}	Uuid	The user ID.

This operation does not accept a request body.

6.5.6.2. Response

Example 6.36. List roles for domain user: 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
  }
}
```

```
    }  
}
```

6.5.7. Grant role to domain user

Method	URI	Description
PUT	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Grants a role to a domain user.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.7.1. Request

This table shows the header parameters for the grant role to domain user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the grant role to domain user 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.

6.5.8. Check role for domain user

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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.8.1. Request

This table shows the header parameters for the check role for domain user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the check role for domain user 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.

6.5.9. Revoke role from domain user

Method	URI	Description
DELETE	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Revokes a role from a domain user.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.9.1. Request

This table shows the header parameters for the revoke role from domain user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the revoke role from domain user 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.

6.5.10. List roles for domain group

Method	URI	Description
GET	/v3/domains/{domain_id}/groups/{group_id}/roles	Lists roles for a domain group.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.10.1. Request

This table shows the header parameters for the list roles for domain group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the list roles for domain group request:

Name	Type	Description
{domain_id}	Uuid	The domain ID.
{group_id}	Uuid	The group ID.

This operation does not accept a request body.

6.5.10.2. Response

Example 6.37. List roles for domain group: 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
  }
}
```

```
    }  
}
```

6.5.11. Grant role to domain group

Method	URI	Description
PUT	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Grants a role to a domain group.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.11.1. Request

This table shows the header parameters for the grant role to domain group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the grant role to domain group 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.

6.5.12. Check role for domain group

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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.12.1. Request

This table shows the header parameters for the check role for domain group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the check role for domain group 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.

6.5.13. Revoke role from domain group

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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.5.13.1. Request

This table shows the header parameters for the revoke role from domain group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the revoke role from domain group 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.

6.6. Projects

Manages projects.

Method	URI	Description
POST	/v3/projects	Creates a project.
GET	/v3/projects{?domain_id, parent_id, name, enabled}	Lists projects.
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.
GET	/v3/projects/{project_id}/users/{user_id}/roles	Lists roles for a user in a project.
PUT	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Grants a role to a user in a project.
HEAD	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Validates that a user has a role in a project.
DELETE	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Revokes a role from a project user.
GET	/v3/projects/{project_id}/groups/{group_id}/roles	Lists roles for a project group.

Method	URI	Description
PUT	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Grants a role to a project group.
HEAD	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Validates that a project group has a role.
DELETE	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Revokes a role from a project group.

6.6.1. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.1.1. Request

This table shows the header parameters for the create project request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 6.38. Create project: JSON request

```
{
  "project": {
    "description": "My new project",
    "domain_id": "1789d1",
    "parent_id": "123c56",
    "enabled": true,
    "name": "myNewProject"
  }
}
```

6.6.1.2. Response

Example 6.39. Create project: JSON response

```
{
  "project": {
    "domain_id": "1789d1",
    "parent_id": "123c56",
    "enabled": true,
    "id": "456789",
    "links": {
      "self": "http://identity:35357/v3/projects/456789"
    },
    "name": "myNewProject"
  }
}
```

6.6.2. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.2.1. Request

This table shows the header parameters for the list projects request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
domain_id	String <i>(Optional)</i>	Filters on domain_id.
parent_id	String <i>(Optional)</i>	Filters on parent_id.
name	String <i>(Optional)</i>	Filters on a name.
enabled	String <i>(Optional)</i>	Filters on enabled or disabled roles. Values are true or false.

This operation does not accept a request body.

6.6.2.2. Response

Example 6.40. List projects: JSON response

```
{
  "projects": [
    {
      "domain_id": "1789d1",
      "parent_id": "123c56",
      "enabled": true,
      "id": "263fd9",
      "links": {
        "self": "https://identity:35357/v3/projects/263fd9"
      },
      "name": "Test Group"
    }
  ]
}
```

```
{  
    "domain_id": "1789d1",  
    "parent_id": "123c56",  
    "enabled": true,  
    "id": "50ef01",  
    "links": {  
        "self": "https://identity:35357/v3/projects/50ef01"  
    },  
    "name": "Build Group"  
}  
,  
"links": {  
    "self": "https://identity:35357/v3/users/9feld3/projects",  
    "previous": null,  
    "next": null  
}  
}
```

6.6.3. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.3.1. Request

This table shows the header parameters for the show project details request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{project_id}	String	The project ID.

This operation does not accept a request body.

6.6.3.2. Response

Example 6.41. Show project details: JSON response

```
{
  "project": {
    "domain_id": "1789d1",
    "parent_id": "123c56",
    "enabled": true,
    "id": "456789",
    "links": {
      "self": "http://identity:35357/v3/projects/456789"
    },
    "name": "myNewProject"
  }
}
```

6.6.4. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.4.1. Request

This table shows the header parameters for the update project request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{project_id}	String	The project ID.

Example 6.42. Update project: JSON request

```
{
  "project": {
    "description": "My updated project",
    "domain_id": "1789d1",
    "parent_id": "123c56",
    "enabled": true,
    "name": "myUpdatedProject"
  }
}
```

6.6.4.2. Response

Example 6.43. Update project: JSON response

```
{
  "project": {
    "description": "My updated project",
    "domain_id": "1789d1",
    "parent_id": "123c56",
    "enabled": true,
    "id": "263fd9",
    "links": {
      "self": "http://identity:35357/v3/projects/263fd9"
    },
    "name": "myUpdatedProject"
  }
}
```


6.6.5. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.5.1. Request

This table shows the header parameters for the delete project request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{project_id}	String	The project ID.

This operation does not accept a request body.

6.6.6. List roles for project user

Method	URI	Description
GET	/v3/projects/{project_id}/users/{user_id}/roles	Lists roles for a user in a project.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.6.1. Request

This table shows the header parameters for the list roles for project user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the list roles for project user request:

Name	Type	Description
{project_id}	String	The project ID.
{user_id}	Uuid	The user ID.

This operation does not accept a request body.

6.6.6.2. Response

Example 6.44. List roles for project user: 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/users/313233/roles",
    "previous": null,
    "next": null
  }
}
```

```
    }  
}
```

6.6.7. Grant role to project user

Method	URI	Description
PUT	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Grants a role to a user in a project.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.7.1. Request

This table shows the header parameters for the grant role to project user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the grant role to project user request:

Name	Type	Description
{project_id}	String	The project ID.
{user_id}	Uuid	The user ID.
{role_id}	Uuid	The role ID.

This operation does not accept a request body.

6.6.8. Check role for project user

Method	URI	Description
HEAD	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Validates that a user has a role in a project.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.8.1. Request

This table shows the header parameters for the check role for project user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the check role for project user request:

Name	Type	Description
{project_id}	String	The project ID.
{user_id}	Uuid	The user ID.
{role_id}	Uuid	The role ID.

This operation does not accept a request body.

6.6.9. Revoke role from project user

Method	URI	Description
DELETE	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Revokes a role from a project user.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.9.1. Request

This table shows the header parameters for the revoke role from project user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the revoke role from project user request:

Name	Type	Description
{project_id}	String	The project ID.
{user_id}	Uuid	The user ID.
{role_id}	Uuid	The role ID.

This operation does not accept a request body.

6.6.10. List roles for project group

Method	URI	Description
GET	/v3/projects/{project_id}/groups/{group_id}/roles	Lists roles for a project group.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.10.1. Request

This table shows the header parameters for the list roles for project group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the list roles for project group request:

Name	Type	Description
{project_id}	String	The project ID.
{group_id}	Uuid	The group ID.

This operation does not accept a request body.

6.6.10.2. Response

Example 6.45. List roles for project group: 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
  }
}
```

```
        "next": null
    }
}
```

6.6.11. Grant role to project group

Method	URI	Description
PUT	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Grants a role to a project group.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.11.1. Request

This table shows the header parameters for the grant role to project group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the grant role to project group request:

Name	Type	Description
{project_id}	String	The project ID.
{group_id}	Uuid	The group ID.
{role_id}	Uuid	The role ID.

This operation does not accept a request body.

6.6.12. Check role for project group

Method	URI	Description
HEAD	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Validates that a project group has a role.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.12.1. Request

This table shows the header parameters for the check role for project group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the check role for project group request:

Name	Type	Description
{project_id}	String	The project ID.
{group_id}	Uuid	The group ID.
{role_id}	Uuid	The role ID.

This operation does not accept a request body.

6.6.13. Revoke role from project group

Method	URI	Description
DELETE	/v3/projects/{project_id}/groups/{group_id}/roles/{role_id}	Revokes a role from a project group.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.6.13.1. Request

This table shows the header parameters for the revoke role from project group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the revoke role from project group request:

Name	Type	Description
{project_id}	String	The project ID.
{group_id}	Uuid	The group ID.
{role_id}	Uuid	The role ID.

This operation does not accept a request body.

6.7. Users

Manages users.

Method	URI	Description
POST	/v3/users	Creates a user.
GET	/v3/users{?domain_id,name(enabled)}	Lists users.
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.
GET	/v3/users/{user_id}/groups	Lists groups for a user.
GET	/v3/users/{user_id}/projects	List projects for a user.

6.7.1. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.7.1.1. Request

This table shows the header parameters for the create user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 6.46. Create user: JSON request

```
{
  "user": {
    "default_project_id": "263fd9",
    "description": "James Doe's user",
    "domain_id": "1789d1",
    "email": "jdoe@example.com",
    "enabled": true,
    "name": "James Doe",
    "password": "secretsecret"
  }
}
```

6.7.1.2. Response

Example 6.47. Create user: JSON response

```
{
  "user": {
    "default_project_id": "263fd9",
    "description": "James Doe's user",
    "domain_id": "1789d1",
    "email": "jdoe@example.com",
    "enabled": true,
    "id": "ff4e51",
    "links": {
      "self": "https://identity:35357/v3/users/ff4e51"
    },
    "name": "James Doe"
  }
}
```

6.7.2. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.7.2.1. Request

This table shows the header parameters for the list users request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
domain_id	String <i>(Optional)</i>	Filters on domain_id.
name	String <i>(Optional)</i>	Filters on a name.
enabled	String <i>(Optional)</i>	Filters on enabled or disabled roles. Values are true or false.

This operation does not accept a request body.

6.7.2.2. Response

Example 6.48. List users: JSON response

```
{
  "users": [
    {
      "default_project_id": "263fd9",
      "description": "Admin user",
      "domain_id": "1789d1",
      "email": "admin@example.com",
      "enabled": true,
      "id": "0ca8f6",
      "links": {
        "self": "https://identity:35357/v3/users/0ca8f6"
      },
      "name": "admin"
    },
    {
      "default_project_id": "263fd9",
      "description": "User 1",
      "domain_id": "1789d2",
      "email": "user1@example.com",
      "enabled": false,
      "id": "123456",
      "links": {
        "self": "https://identity:35357/v3/users/123456"
      },
      "name": "User 1"
    }
  ]
}
```

```
        "description": "John Smith's user",
        "domain_id": "1789d1",
        "email": "jsmith@example.com",
        "enabled": true,
        "id": "9feld3",
        "links": {
            "self": "https://identity:35357/v3/users/9feld3"
        },
        "name": "jsmith"
    }
],
"links": {
    "self": "http://identity:35357/v3/users",
    "previous": null,
    "next": null
}
}
```

6.7.3. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.7.3.1. Request

This table shows the header parameters for the show user details request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.7.3.2. Response

Example 6.49. 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": "9feld3",
    "links": {
      "self": "https://identity:35357/v3/users/9feld3"
    },
    "name": "jsmith"
  }
}
```

6.7.4. 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 allow this functionality, this call might return the HTTP Not Implemented (501) status code.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.7.4.1. Request

This table shows the header parameters for the update user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{user_id}	Uuid	The user ID.

Example 6.50. Update user: JSON request

```
{
  "user": {
    "default_project_id": "263fd9",
    "description": "James Doe's user",
    "email": "jamesdoe@example.com",
    "enabled": true
  }
}
```

6.7.4.2. Response

Example 6.51. 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"
    }
```

6.7.5. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.7.5.1. Request

This table shows the header parameters for the delete user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.7.6. List groups for user

Method	URI	Description
GET	/v3/users/{user_id}/groups	Lists groups for a user.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.7.6.1. Request

This table shows the header parameters for the list groups for user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{user_id}	Uuid	The user ID.

This operation does not accept a request body.

6.7.6.2. Response

Example 6.52. List groups for user: 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/9feld3/groups",
        "previous": null,
        "next": null
    }
}
```

6.7.7. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.7.7.1. Request

This table shows the header parameters for the list projects for user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.7.7.2. Response

Example 6.53. 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
    }
}
```

6.8. Groups

Manages groups.

Method	URI	Description
POST	/v3/groups	Creates a group.
GET	/v3/groups{?domain_id}	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{?name, domain_id,description,name,enabled}	Lists users in a group.
PUT	/v3/groups/{group_id}/users/{user_id}	Assigns a user to a group.
DELETE	/v3/groups/{group_id}/users/{user_id}	Removes a user from a group.
HEAD	/v3/groups/{group_id}/users/{user_id}	Validates that a user is in a group.

6.8.1. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.8.1.1. Request

This table shows the header parameters for the create group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 6.54. Create group: JSON request

```
{
  "group": {
    "description": "Developers cleared for work on secret projects",
    "domain_id": "161718",
    "name": "Secure Developers"
  }
}
```

6.8.1.2. Response

Example 6.55. Create group: JSON response

```
{
  "group": {
    "description": "Developers cleared for work on secret projects",
    "domain_id": "161718",
    "id": "101112",
    "links": {
      "self": "http://identity:35357/v3/groups/101112"
    },
    "name": "Secure Developers"
  }
}
```

6.8.2. List groups

Method	URI	Description
GET	/v3/groups{?domain_id}	Lists groups.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.8.2.1. Request

This table shows the header parameters for the list groups request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
domain_id	String <i>(Optional)</i>	Filters on domain_id.

This operation does not accept a request body.

6.8.2.2. Response

Example 6.56. List groups: JSON response

```
{
  "groups": [
    {
      "description": "Developers cleared for work on all general projects",
      "domain_id": "161718",
      "id": "101112",
      "links": {
        "self": "http://identity:35357/v3/groups/101112"
      },
      "name": "Developers"
    },
    {
      "description": "Developers cleared for work on secret projects",
      "domain_id": "161718",
      "id": "101113",
      "links": {
        "self": "http://identity:35357/v3/groups/101113"
      },
      "name": "Secure Developers"
    }
  ]
}
```

```
        "description": "Testers cleared for work on all general projects",
        "domain_id": "161718",
        "id": "101114",
        "links": [
            "self": "http://identity:35357/v3/groups/101114"
        ],
        "name": "Testers"
    },
],
"links": [
    "self": "http://identity:35357/v3/groups",
    "previous": null,
    "next": null
}
}
```

6.8.3. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.8.3.1. Request

This table shows the header parameters for the show group details request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.8.3.2. Response

Example 6.57. Show group details: JSON response

```
{
  "group": {
    "description": "Developers cleared for work on secret projects",
    "domain_id": "161718",
    "id": "101112",
    "links": {
      "self": "http://identity:35357/v3/groups/101112"
    },
    "name": "Secure Developers"
  }
}
```

6.8.4. Update group

Method	URI	Description
PATCH	/v3/groups/{group_id}	Updates a group.

If the back-end driver does not allow this functionality, this operation might return the HTTP Not Implemented (501) status code.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.8.4.1. Request

This table shows the header parameters for the update group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{group_id}	Uuid	The group ID.

Example 6.58. Update group: JSON request

```
{
  "group": {
    "description": "my updated group",
    "name": "myUpdatedGroup"
  }
}
```

6.8.4.2. Response

Example 6.59. Update group: JSON response

```
{
  "group": {
    "description": "my updated group",
    "domain_id": "161718",
    "id": "101112",
    "links": {
      "self": "http://identity:35357/v3/groups/101112"
    },
    "name": "myUpdatedGroup"
  }
}
```

6.8.5. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.8.5.1. Request

This table shows the header parameters for the delete group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.8.6. List users in group

Method	URI	Description
GET	/v3/groups/{group_id}/users{?name, domain_id,description,name(enabled)}	Lists users in a group.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.8.6.1. Request

This table shows the header parameters for the list users in group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the list users in group request:

Name	Type	Description
{group_id}	Uuid	The group ID.

This table shows the query parameters for the list users in group request:

Name	Type	Description
name	String <i>(Optional)</i>	Filters on a name.
domain_id	String <i>(Optional)</i>	Filters on domain_id.
description	String <i>(Optional)</i>	Filters on a description.
name	String <i>(Optional)</i>	Filters on a name.
enabled	String <i>(Optional)</i>	Filters on enabled or disabled roles. Values are true or false.

This operation does not accept a request body.

6.8.6.2. Response

Example 6.60. List users in group: JSON response

```
{
  "users": [
```

```
{  
    "default_project_id": "414345",  
    "description": "a user",  
    "domain_id": "161718",  
    "email": "user@example.com",  
    "enabled": true,  
    "id": "313233",  
    "links": {  
        "self": "http://identity:35357/v3/users/313233"  
    },  
    "name": "admin"  
},  
{  
    "default_project_id": "414345",  
    "description": "another user",  
    "domain_id": "161718",  

```

6.8.7. Assign user to group

Method	URI	Description
PUT	/v3/groups/{group_id}/users/{user_id}	Assigns a user to a group.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.8.7.1. Request

This table shows the header parameters for the assign user to group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the assign 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.

6.8.8. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.8.8.1. Request

This table shows the header parameters for the remove user from group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.8.9. Check user membership in group

Method	URI	Description
HEAD	/v3/groups/{group_id}/users/{user_id}	Validates that a user is in a group.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.8.9.1. Request

This table shows the header parameters for the check user membership in group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the check user membership in 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.

6.9. Credentials

Manages credentials.

Method	URI	Description
POST	/v3/credentials	Creates a credential.
GET	/v3/credentials{?user_id}	Lists 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.

6.9.1. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.9.1.1. Request

This table shows the header parameters for the create credential request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 6.61. Create credential: JSON request

```
{
  "blob": "{\"access\": \"181920\", \"secret\": \"secretKey\"}",
  "project_id": "456789",
  "type": "ec2",
  "user_id": "616263"
}
```

6.9.1.2. Response

Example 6.62. Create credential: JSON response

```
{
  "blob": "{\"access\": \"181920\", \"secret\": \"secretKey\"}",
  "id": "414243",
  "links": {
    "self": "http://identity:35357/v3/credentials/414243"
  },
  "project_id": "456789",
  "type": "ec2",
  "user_id": "616263"
}
```

6.9.2. List credentials

Method	URI	Description
GET	/v3/credentials{?user_id}	Lists credentials.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.9.2.1. Request

This table shows the header parameters for the list credentials request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
user_id	String <i>(Optional)</i>	Filters on user_id.

This operation does not accept a request body.

6.9.2.2. Response

Example 6.63. List credentials: JSON response

```
[
  {
    "blob": "{\"access\":\"181920\",\"secret\":\"secretKey\"}",
    "id": "414243",
    "links": {
      "self": "http://identity:35357/v3/credentials/414243"
    },
    "project_id": "456789",
    "type": "ec2",
    "user_id": "616263"
  },
  {
    "blob": "{\"access\":\"181920\",\"secret\":\"secretKey\"}",
    "id": "414243",
    "links": {
      "self": "http://identity:35357/v3/credentials/414243"
    },
    "project_id": "456789",
    "type": "ec2",
    "user_id": "616263"
  }
]
```


6.9.3. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.9.3.1. Request

This table shows the header parameters for the show credential details request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{credential_id}	Uuid	The credential ID.

This operation does not accept a request body.

6.9.3.2. Response

Example 6.64. Show credential details: JSON response

```
{
  "blob": "{\"access\": \"181920\", \"secret\": \"secretKey\"}",
  "id": "414243",
  "links": {
    "self": "http://identity:35357/v3/credentials/414243"
  },
  "project_id": "456789",
  "type": "ec2",
  "user_id": "616263"
}
```

6.9.4. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.9.4.1. Request

This table shows the header parameters for the update credential request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{credential_id}	Uuid	The credential ID.

Example 6.65. Update credential: JSON request

```
{
  "blob": "{\"access\": \"181920\", \"secret\": \"secretKey\"} ,
  "project_id": "456789",
  "type": "ec2",
  "user_id": "616263"
}
```

6.9.4.2. Response

Example 6.66. Update credential: JSON response

```
{
  "blob": "{\"access\": \"181920\", \"secret\": \"secretKey\"} ,
  "id": "414243",
  "links": {
    "self": "http://identity:35357/v3/credentials/414243"
  },
  "project_id": "456789",
  "type": "ec2",
  "user_id": "616263"
}
```

6.9.5. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.9.5.1. Request

This table shows the header parameters for the delete credential request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{credential_id}	Uuid	The credential ID.

This operation does not accept a request body.

6.10. Roles

Manages roles.

Method	URI	Description
POST	/v3/roles	Creates a role.
GET	/v3/roles{?name}	Lists roles.
GET	/v3/role_assignments{?group.id, role.id, scope.domain.id, scope.project.id, user.id, effective}	Lists role assignments.
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.

6.10.1. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.10.1.1. Request

This table shows the header parameters for the create role request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 6.67. Create role: JSON request

```
{  
    "role": {  
        "name": "developer"  
    }  
}
```

6.10.1.2. Response

Example 6.68. Create role: JSON response

```
{  
    "role": {  
        "id": "123456",  
        "links": {  
            "self": "http://identity:35357/v3/roles/123456"  
        },  
        "name": "developer"  
    }  
}
```

6.10.2. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.10.2.1. Request

This table shows the header parameters for the list roles request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
name	String <i>(Optional)</i>	Filters on a name.

This operation does not accept a request body.

6.10.2.2. Response

Example 6.69. List roles: 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/roles",
    "previous": null,
    "next": null
  }
}
```

}

6.10.3. 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.

Because the role assignments list can be long, use the query parameters to filter the list.

Some typical examples are:

GET /role_assignments?user.id={user_id} lists role assignments for a user.

GET /role_assignments?scope.project.id={project_id} lists 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. 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. You can use the other query parameters with the `effective` parameter.

For example, to determine what a user can actually do, issue the GET /role_assignments?user.id={user_id}&effective request.

To return the equivalent set of role assignments that would be included in the token response of a project-scoped token, issue the GET /role_assignments?user.id={user_id}&scope.project.id={project_id}&effective request.

In the response, the `links` entity 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.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.10.3.1. Request

This table shows the header parameters for the list role assignments request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This operation does not accept a request body.

6.10.3.2. Response

Example 6.70. List role assignments: JSON response

```
GET /role_assignments?user.id={user_id}&scope.project.id={project_id}&effective
```

Example 6.71. 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
    }
}
```

Example 6.72. List role assignments: JSON response

```
GET /role_assignments?user.id={user_id}&scope.project.id={project_id}&effective
```

Example 6.73. 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"  
            }  
        },  
        {  
            "links": {  
                "assignment": "http://identity:35357/v3/projects/456789/  
groups/101112/roles/123456",  
                "membership": "http://identity:35357/v3/groups/101112/users/  
313233"  
            },  
            "role": {  
                "id": "123456"  
            },  
            "scope": {  
                "project": {  
                    "id": "456789"  
                }  
            },  
            "user": {  
                "id": "313234"  
            }  
        }  
    ],  
    "links": {  
        "self": "http://identity:35357/v3/role_assignments?effective",  
        "previous": null,  
        "next": null  
    }  
}
```

6.10.4. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.10.4.1. Request

This table shows the header parameters for the show role details request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.10.4.2. Response

Example 6.74. Show role details: JSON response

```
{
  "id": "123456",
  "links": {
    "self": "http://identity:35357/v3/roles/131415"
  },
  "name": "developer"
}
```

6.10.5. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.10.5.1. Request

This table shows the header parameters for the update role request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{role_id}	Uuid	The role ID.

Example 6.75. Update role: JSON request

```
{
  "role": {
    "name": "a new role name"
  }
}
```

6.10.5.2. Response

Example 6.76. Update role: JSON response

```
{
  "id": "123456",
  "links": {
    "self": "http://identity:35357/v3/roles/131415"
  },
  "name": "developer"
}
```

6.10.6. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.10.6.1. Request

This table shows the header parameters for the delete role request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.11. Policies

Manages policies.

You can encode policy rule sets into a blob to be consumed by remote services. To do so, set type to application/json and specify policy rules as JSON strings in a blob. For example:

```
"blob": {
    "default": false
}
```

Or:

```
"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.

Method	URI	Description
DELETE	/v3/policies/{policy_id}	Deletes a policy.

6.11.1. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.11.1.1. Request

This table shows the header parameters for the create policy request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 6.77. Create policy: JSON request

```
{
  "policy": {
    "blob": "{ 'foobar_user': 'role:compute-user' }",
    "project_id": "0426ac1e48f642ef9544c2251e07e261",
    "type": "application/json",
    "user_id": "0ffd248c55b443eaac5253b4e9cbf9b5"
  }
}
```

6.11.1.2. Response

Example 6.78. Create policy: JSON response

```
{
  "policy": {
    "user_id": "0ffd248c55b443eaac5253b4e9cbf9b5",
    "links": {
      "self": "http://identity:35357/v3/policies/13c92821e4c4476a878d3aae7444f52f"
    },
    "blob": "{ 'foobar_user' : 'role:compute-user' }",
    "project_id": "0426ac1e48f642ef9544c2251e07e261",
    "type": "application/json",
    "id": "13c92821e4c4476a878d3aae7444f52f"
  }
}
```

6.11.2. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.11.2.1. Request

This table shows the header parameters for the list policies request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
type	String <i>(Optional)</i>	Filters by service type. Service types include compute, ec2, image, and identity.

This operation does not accept a request body.

6.11.2.2. Response

Example 6.79. List policies: JSON response

```
[  
  {  
    "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"  
  }]
```

```
        "links": {
            "self": "http://identity:35357/v3/policies/717274"
        },
        "project_id": "456789",
        "type": "application/json",
        "user_id": "616263"
    }
]
```

6.11.3. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.11.3.1. Request

This table shows the header parameters for the show policy details request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.11.3.2. Response

Example 6.80. Show policy details: JSON response

```
{
  "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"
}
```

6.11.4. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.11.4.1. Request

This table shows the header parameters for the update policy request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{policy_id}	Uuid	The policy ID.

Example 6.81. Update policy: JSON request

```
{
  "blob": {
    "foobar_user": [
      "role:compute-user"
    ],
    "project_id": "456789",
    "type": "application/json",
    "user_id": "616263"
  }
}
```

6.11.4.2. Response

Example 6.82. Update policy: JSON response

```
{
  "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"  
    }
```

6.11.5. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.11.5.1. Request

This table shows the header parameters for the delete policy request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.12. Regions

Manages regions.

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.

Method	URI	Description
GET	/v3/regions{?parent_region_id}	Lists regions.
POST	/v3/regions	Creates a region.
PUT	/v3/regions/{user_defined_region_id}	Creates a region with a user-defined region ID.
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.

6.12.1. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.12.1.1. Request

This table shows the header parameters for the list regions request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
parent_region_id	String <i>(Optional)</i>	Filters the results by a parent region, by ID.

This operation does not accept a request body.

6.12.1.2. Response

Example 6.83. List regions: JSON response

```
{
  "regions": [
    {
      "description": "US East Region",
      "id": "us-east",
      "links": {
        "self": "https://identity:35357/v3/regions/us-east",
        "child_regions": "https://identity:35357/v3/regions?
parent_region_id=us-east"
      },
      "parent_region_id": "us-east-coast"
    }
  ],
  "links": {
    "self": "https://identity:35357/v3/regions",
    "previous": null,
    "next": null
  }
}
```

6.12.2. Create region

Method	URI	Description
POST	/v3/regions	Creates a region.

If you try to add a region with a parent region ID that does not exist, this call returns the HTTP Not Found (404) status code.

If you try to add a region with a parent region ID that would form a circular relationship, this call returns the HTTP Conflict (409) status code.

The user-defined region ID must be unique to the OpenStack deployment. Otherwise, this call returns the HTTP Conflict (409) status code.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.12.2.1. Request

This table shows the header parameters for the create region request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 6.84. Create region: JSON request

```
{
    "region": {
        "description": "US West Subregion 1",
        "parent_region_id": "829551"
    }
}
```

6.12.2.2. Response

Example 6.85. Create region: JSON response

```
{
    "region": {
        "description": "US West Subregion 1",
        "id": "8ebd7f",
        "links": {
            "self": "https://identity:35357/v3/regions/8ebd7f",
            "child_regions": "https://identity:35357/v3/regions?
parent_region_id=8ebd7f"
        },
        "parent_region_id": "829551"
    }
}
```


6.12.3. Create region with user-defined ID

Method	URI	Description
PUT	/v3/regions/{user_defined_region_id}	Creates a region with a user-defined region ID.

You must URL-encode the user-defined region ID if it contains characters that are not allowed in an URI.

If you try to add a region with a parent region ID that does not exist, this call returns the **HTTP Not Found (404)** status code.

If you try to add a region with a parent region ID that would form a circular relationship, this call returns the **HTTP Conflict (409)** status code.

The user-defined region ID must be unique to the OpenStack deployment. Otherwise, this call returns the **HTTP Conflict (409)** status code.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.12.3.1. Request

This table shows the header parameters for the create region with user-defined id request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the create region with user-defined id request:

Name	Type	Description
{user_defined_region_id}	String	<p>A user-defined region ID.</p> <p>You must URL-encode the user-defined region ID if it contains characters that are not allowed in an URI.</p> <p>The user-defined region ID must be unique to the OpenStack deployment. Otherwise, the call returns the HTTP Conflict (409) status code.</p>

Example 6.86. Create region with user-defined ID: JSON request

```
{
    "region": {
        "description": "US West Subregion 1",
        "parent_region_id": "829551"
    }
}
```

6.12.3.2. Response

Example 6.87. Create region with user-defined ID: JSON response

```
{  
    "region": {  
        "description": "US West Subregion 1",  
        "id": "8ebd7f",  
        "links": {  
            "self": "https://identity:35357/v3/regions/8ebd7f",  
            "child_regions": "https://identity:35357/v3/regions?  
parent_region_id=8ebd7f"  
        },  
        "parent_region_id": "829551"  
    }  
}
```

6.12.4. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.12.4.1. Request

This table shows the header parameters for the show region details request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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.

6.12.4.2. Response

Example 6.88. Show region details: JSON response

```
{
  "region": {
    "description": "US Southwest Region",
    "id": "us-southwest",
    "links": {
      "self": "https://identity:35357/v3/regions/us-southwest",
      "child_regions": "http://identity:35357/v3/regions?
parent_region_id=us-southwest"
    },
    "parent_region_id": "us-west-coast"
  }
}
```

6.12.5. Update region

Method	URI	Description
PATCH	/v3/regions/{region_id}	Updates a region.

If you try to update a region with a parent region ID that does not exist, this call returns the HTTP Not Found (404) status code.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.12.5.1. Request

This table shows the header parameters for the update region request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{region_id}	Uuid	The region ID.

Example 6.89. Update region: JSON request

```
{
  "region": {
    "description": "US Southwest Subregion",
    "parent_region_id": "us-southwest"
  }
}
```

6.12.5.2. Response

Example 6.90. Update region: JSON response

```
{
  "region": {
    "description": "US Southwest Subregion",
    "id": "us-southwest-1",
    "links": {
      "self": "https://identity:35357/v3/regions/us-southwest-1",
      "child_regions": "https://identity:35357/v3/regions?
parent_region_id=us-southwest-1"
    },
    "parent_region_id": "us-southwest"
  }
}
```

6.12.6. Delete region

Method	URI	Description
DELETE	/v3/regions/{region_id}	Deletes a region.



Warning

If you try to delete a region that has child regions, this call returns the HTTP Conflict (409) status code.

Normal response codes: 204

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

6.12.6.1. Request

This table shows the header parameters for the delete region request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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. Identity API v3 extensions (CURRENT)

This page describes these Identity API v3 extensions:

- OS-INHERIT extension (OS-INHERIT)
- Key Distribution Server (KDS) extension (OS-KDS)
- OAuth extension (OS-OAUTH1)
- Trust extension (OS-TRUST)

Method	URI	Description
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 inherited project roles for a user on a project.
PUT	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a user on 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 <code>inherited_to_projects</code> flag on 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 on a project.
GET	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/inherited_to_projects	Lists the inherited project roles for a group on a project.
PUT	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a group on 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 <code>inherited_to_projects</code> flag on 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 on a project.
GET	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/inherited_to_projects	Lists the inherited project roles on a domain for a user.
PUT	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a user on projects owned by a domain.
HEAD	/v3/OS-INHERIT/domains/{domain_id}/users/	Checks whether a user has an inherited project role on a domain.

Method	URI	Description
	{user_id}/roles/{role_id}/inherited_to_projects	
DELETE	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited project role from a user on a domain.
GET	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/inherited_to_projects	Lists the inherited project roles on a domain for a group.
PUT	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a group on 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 on 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 on a domain.
Key Distribution Server (KDS) extension (OS-KDS)		
POST	/v1/keys/{name}	Creates a long-term key in the KDS.
DELETE	/v1/keys/{name}	Deletes a long-term key from the KDS.
POST	/v1/tickets	Generates a ticket to facilitate messaging between a source and destination.
GET	/v1/groups	Gets the key for a group in the KDS.
POST	/v1/groups/{name}	Creates a group in the KDS.
DELETE	/v1/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.

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.
POST	/v3/auth/tokens	Enables a consumer to get an Identity Service authentication token.
Trust extension (OS-TRUST)		
POST	/v3/OS-TRUST/trust	Creates a trust.

7.1. OS-INHERIT extension (OS-INHERIT)

Enables projects to inherit role assignments from either their owning domain or projects that are higher in the hierarchy.

The OS-INHERIT extension was extended in API v3.4 to allow inheritance from projects in addition to inheritance from 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 inherited project roles for a user on a project.
PUT	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a user on 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 <code>inherited_to_projects</code> flag on 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 on a project.
GET	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/inherited_to_projects	Lists the inherited project roles for a group on a project.
PUT	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a group on 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 <code>inherited_to_projects</code> flag on 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 on a project.
GET	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/inherited_to_projects	Lists the inherited project roles on a domain for a user.

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 on 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 on 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 on a domain.
GET	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/inherited_to_projects	Lists the inherited project roles on a domain for a group.
PUT	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/{role_id}/inherited_to_projects	Assigns a role to a group on 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 on 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 on a domain.

7.1.1. 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 return 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.

An additional `scope.OS-INHERIT:inherited_to` query filter enables filtering based on role assignments that are inherited. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

7.1.1.1. Request

This operation does not accept a request body.

7.1.1.2. Response

Example 7.1. 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 7.2. 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  
    }  
}
```

7.1.2. List roles for user

Method	URI	Description
GET	/v3/OS-INHERIT/projects/{project_id}/users/{user_id}/roles/inherited_to_projects	Lists the inherited project roles for a user on a project.

The list shows only roles inherited from the parent project.

Normal response codes: 200

7.1.2.1. Request

This table shows the URI parameters for the list roles for user request:

Name	Type	Description
{project_id}	Uuid	The ID of the project.
{user_id}	Uuid	The ID of the user.

This operation does not accept a request body.

7.1.2.2. Response

Example 7.3. 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
  }
}
```

7.1.3. 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 on projects in a subtree.

The inherited role assignment is anchored to a project and applied 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 on the same project.

Normal response codes: 204

7.1.3.1. Request

This table shows the URI parameters for the assign role to user request:

Name	Type	Description
{project_id}	Uuid	The ID of the project.
{user_id}	Uuid	The ID of the user.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.1.4. 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 <code>inherited_to_projects</code> flag on a project.

Normal response codes: 200

7.1.4.1. Request

This table shows the URI parameters for the check role for user request:

Name	Type	Description
{project_id}	Uuid	The ID of the project.
{user_id}	Uuid	The ID of the user.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.1.5. 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 on a project.

Normal response codes: 204

7.1.5.1. Request

This table shows the URI parameters for the revoke role from user request:

Name	Type	Description
{project_id}	Uuid	The ID of the project.
{user_id}	Uuid	The ID of the user.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.1.6. List roles for group

Method	URI	Description
GET	/v3/OS-INHERIT/projects/{project_id}/groups/{group_id}/roles/inherited_to_projects	Lists the inherited project roles for a group on a project.

The list shows only roles inherited from the parent project.

Normal response codes: 200

7.1.6.1. Request

This table shows the URI parameters for the list roles for group request:

Name	Type	Description
{project_id}	Uuid	The ID of the project.
{group_id}	Uuid	The ID of the group.

This operation does not accept a request body.

7.1.6.2. Response

Example 7.4. 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
  }
}
```

7.1.7. 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 on projects in a subtree.

The inherited role assignment is anchored to a project and applied 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 on the same project.

Normal response codes: 204

7.1.7.1. Request

This table shows the URI parameters for the assign role to group request:

Name	Type	Description
{project_id}	Uuid	The ID of the project.
{group_id}	Uuid	The ID of the group.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.1.8. 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 <code>inherited_to_projects</code> flag on a project.

Normal response codes: 200

7.1.8.1. Request

This table shows the URI parameters for the check role for group request:

Name	Type	Description
{project_id}	Uuid	The ID of the project.
{group_id}	Uuid	The ID of the group.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.1.9. 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 on a project.

Normal response codes: 204

7.1.9.1. Request

This table shows the URI parameters for the revoke role from group request:

Name	Type	Description
{project_id}	Uuid	The ID of the project.
{group_id}	Uuid	The ID of the group.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.1.10. List project roles for user on domain

Method	URI	Description
GET	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/inherited_to_projects	Lists the inherited project roles on a domain for a user.

The list shows only roles inherited from the parent project within the domain.

Normal response codes: 200

7.1.10.1. Request

This table shows the URI parameters for the list project roles for user on domain request:

Name	Type	Description
{domain_id}	Uuid	The ID of the domain.
{user_id}	Uuid	The ID of the user.

This operation does not accept a request body.

7.1.10.2. Response

Example 7.5. List project roles for user on 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
  }
}
```

7.1.11. 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 on projects owned by a domain.

The inherited role is applied to only the owned projects, both existing and future, and does not appear as a role in a domain-scoped token.

Normal response codes: 204

7.1.11.1. 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 ID of the domain.
{user_id}	Uuid	The ID of the user.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.1.12. Check project role for user on domain

Method	URI	Description
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 on a domain.

Normal response codes: 200

7.1.12.1. Request

This table shows the URI parameters for the check project role for user on domain request:

Name	Type	Description
{domain_id}	Uuid	The ID of the domain.
{user_id}	Uuid	The ID of the user.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.1.13. Revoke project role from user on domain

Method	URI	Description
DELETE	/v3/OS-INHERIT/domains/{domain_id}/users/{user_id}/roles/{role_id}/inherited_to_projects	Revokes an inherited project role from a user on a domain.

Normal response codes: 204

7.1.13.1. Request

This table shows the URI parameters for the revoke project role from user on domain request:

Name	Type	Description
{domain_id}	Uuid	The ID of the domain.
{user_id}	Uuid	The ID of the user.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.1.14. List project roles for group on domain

Method	URI	Description
GET	/v3/OS-INHERIT/domains/{domain_id}/groups/{group_id}/roles/inherited_to_projects	Lists the inherited project roles on a domain for a group.

The list shows only roles inherited from the parent project within the domain.

Normal response codes: 200

7.1.14.1. Request

This table shows the URI parameters for the list project roles for group on domain request:

Name	Type	Description
{domain_id}	Uuid	The ID of the domain.
{group_id}	Uuid	The ID of the group.

This operation does not accept a request body.

7.1.14.2. Response

Example 7.6. List project roles for group on 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
    }
}
```

7.1.15. Assign role to group on 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 on projects owned by a domain.

The inherited role is applied to only the owned projects, both existing and future, and does not appear as a role in a domain-scoped token.

Normal response codes: 204

7.1.15.1. Request

This table shows the URI parameters for the assign role to group on domain projects request:

Name	Type	Description
{domain_id}	Uuid	The ID of the domain.
{group_id}	Uuid	The ID of the group.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.1.16. Check project role for group on domain

Method	URI	Description
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 on a domain.

Normal response codes: 200

7.1.16.1. Request

This table shows the URI parameters for the check project role for group on domain request:

Name	Type	Description
{domain_id}	Uuid	The ID of the domain.
{group_id}	Uuid	The ID of the group.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.1.17. Revoke project role from group on 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 on a domain.

Normal response codes: 204

7.1.17.1. Request

This table shows the URI parameters for the revoke project role from group on domain request:

Name	Type	Description
{domain_id}	Uuid	The ID of the domain.
{group_id}	Uuid	The ID of the group.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.2. 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 to be 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	/v1/keys/{name}	Creates a long-term key in the KDS.
DELETE	/v1/keys/{name}	Deletes a long-term key from the KDS.
POST	/v1/tickets	Generates a ticket to facilitate messaging between a source and destination.

Method	URI	Description
GET	/v1/groups	Gets the key for a group in the KDS.
POST	/v1/groups/{name}	Creates a group in the KDS.
DELETE	/v1/groups/{name}	Deletes a group from the KDS.

7.2.1. Create key

Method	URI	Description
POST	/v1/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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.2.1.1. 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 7.7. Create key: JSON request

```
{  
    "key": "TXkgcHJ1Y21vdXNzcy4u..."  
}
```

7.2.1.2. Response

Example 7.8. Create key: JSON response

```
{  
    "name": "TXkgcHJ1Y21vdXNzcy4u..." ,  
    "generation": 2  
}
```

7.2.2. Delete key

Method	URI	Description
DELETE	/v1/keys/{name}	Deletes a long-term key from the KDS.

Normal response codes: 204

7.2.2.1. 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.

7.2.3. Generate ticket

Method	URI	Description
POST	/v1/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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.2.3.1. Request

Example 7.9. Generate ticket: JSON request

```
{  
    "metadata": "Zhn8yhasf8hihkf...",  
    "signature": "c2lnbmF0dXJl..."  
}
```

7.2.3.2. Response

Example 7.10. Generate ticket: JSON response

```
{  
    "source": "scheduler.host.example.com",  
    "destination": "compute.host.example.com",  
    "timestamp": "2012-03-26T10:01:01.720000",  
    "nonce": 1234567890  
}
```

7.2.4. Get group key

Method	URI	Description
GET	/v1/groups	Gets 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.2.4.1. Request

This operation does not accept a request body.

7.2.4.2. Response

Example 7.11. Get group key: JSON response

```
{  
    "name": "--group-name--"  
}
```

7.2.5. Create group

Method	URI	Description
POST	/v1/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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.2.5.1. 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.

7.2.5.2. Response

Example 7.12. Create group: JSON response

```
{  
    "name": "--group-name--"  
}
```

7.2.6. Delete group

Method	URI	Description
DELETE	/v1/groups/{name}	Deletes a group from the KDS.

Normal response codes: 204

7.2.6.1. 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.

7.3. OAuth extension (OS-OAUTH1)

Enables users to delegate roles to third-party consumers through the [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. An OAuth-derived token enables admin users to act on behalf of the authorizing user. A request token is a token that the consumer uses to get authorization from the user and exchanges with an OAuth verifier for an access token. The OAuth verifier is a required string that is provided with the corresponding request token in exchange for an access token. An access token is a token that the consumer uses 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-OAUTH1/consumers	Enables a user to create a consumer.

Method	URI	Description
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.

7.3.1. Create consumer

Method	URI	Description
POST	/v3/OS-OAUTH1/consumers	Enables a user to create a consumer.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.3.1.1. Request

Example 7.13. Create consumer: JSON request

```
{  
    "consumer": {  
        "description": "My consumer"  
    }  
}
```

7.3.1.2. Response

Example 7.14. Create consumer: JSON response

```
{  
    "consumer": {  
        "secret": "secretsecret",  
        "description": "My consumer",  
        "id": "7fea2d",  
        "links": {  
            "self": "http://identity:35357/v3/OS-OAUTH1/consumers/7fea2d"  
        }  
    }  
}
```

7.3.2. List consumers

Method	URI	Description
GET	/v3/OS-OAUTH1/consumers	Lists consumers.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

7.3.2.1. Request

This operation does not accept a request body.

7.3.2.2. Response

Example 7.15. List consumers: JSON response

```
{
    "consumers": [
        {
            "id": "0c2a74",
            "links": {
                "self": "http://identity:35357/v3/OS-OAUTH1/consumers/0c2a74"
            }
        },
        {
            "description": "My consumer",
            "id": "7fea2d",
            "links": {
                "self": "http://identity:35357/v3/OS-OAUTH1/consumers/7fea2d"
            }
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "http://identity:35357/v3/OS-OAUTH1/consumers"
    }
}
```

7.3.3. Show consumer details

Method	URI	Description
GET	/v3/OS-OAUTH1/consumers/{consumer_id}	Shows details for a consumer.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

7.3.3.1. Request

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

Name	Type	Description
{consumer_id}	Uuid	The ID of the consumer.

This operation does not accept a request body.

7.3.3.2. Response

Example 7.16. Show consumer details: JSON response

```
{
  "consumer": {
    "id": "7fea2d",
    "description": "My consumer",
    "links": {
      "self": "http://identity:35357/v3/OS-OAUTH1/consumers/7fea2d"
    }
  }
}
```

7.3.4. Update consumer

Method	URI	Description
PATCH	/v3/OS-OAUTH1/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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.3.4.1. Request

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

Name	Type	Description
{consumer_id}	Uuid	The ID of the consumer.

Example 7.17. Update consumer: JSON request

```
{
  "consumer": {
    "description": "My new consumer"
  }
}
```

7.3.4.2. Response

Example 7.18. Update consumer: JSON response

```
{
  "consumer": {
    "description": "My new consumer",
    "id": "7fea2d",
    "links": {
      "self": "http://identity:35357/v3/OS-OAUTH1/consumers/7fea2d"
    }
  }
}
```

7.3.5. Delete consumer

Method	URI	Description
DELETE	/v3/OS-OAUTH1/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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.3.5.1. Request

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

Name	Type	Description
{consumer_id}	Uuid	The ID of the consumer.

This operation does not accept a request body.

7.3.6. Get 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.3.6.1. Request

This operation does not accept a request body.

7.3.6.2. Response

Example 7.19. Get unauthorized request token: application/txtresponse

```
oauth_token=29971f&oauth_token_secret=238eb8&oauth_expires_at=
2013-09-11T06:07:51.501805Z
```

7.3.7. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.3.7.1. Request

This operation does not accept a request body.

7.3.7.2. Response

Example 7.20. Create access token: application/txtresponse

```
oauth_token=accd36&oauth_token_secret=aa47da&oauth_expires_at=
2013-09-11T06:07:51.501805Z
```

7.3.8. List authorized access tokens

Method	URI	Description
GET	/v3/OS-OAUTH1/users/{user_id}/access_tokens	Lists authorized access tokens.

Normal response codes: 200

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.3.8.1. Request

This table shows the URI parameters for the list authorized access tokens request:

Name	Type	Description
{user_id}	Uuid	The ID of the user.

This operation does not accept a request body.

7.3.8.2. Response

Example 7.21. 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-OAUTH1/
access_tokens/6be26a/roles",
        "self": "http://identity:35357/v3/users/ce9e07/OS-OAUTH1/
access_tokens/6be26a"
      },
      "project_id": "b9fca3",
      "authorizing_user_id": "ce9e07"
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "http://identity:35357/v3/users/ce9e07/OS-OAUTH1/
access_tokens"
  }
}
```

7.3.9. Get authorized access token

Method	URI	Description
GET	/v3/OS-OAUTH1/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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.3.9.1. Request

This table shows the URI parameters for the get authorized access token request:

Name	Type	Description
{user_id}	Uuid	The ID of the user.
{access_token_id}	Uuid	The ID of the access token.

This operation does not accept a request body.

7.3.9.2. Response

Example 7.22. 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-OAUTH1/
access_tokens/6be26a/roles",
      "self": "http://identity:35357/v3/users/ce9e07/OS-OAUTH1/
access_tokens/6be26a"
    },
    "project_id": "b9fca3",
    "authorizing_user_id": "ce9e07"
  }
}
```

7.3.10. Revoke access token

Method	URI	Description
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.

Normal response codes: 204

7.3.10.1. Request

This table shows the URI parameters for the revoke access token request:

Name	Type	Description
{user_id}	Uuid	The ID of the user.
{access_token_id}	Uuid	The ID of the access token.

This operation does not accept a request body.

7.3.11. List roles for an access token

Method	URI	Description
GET	/v3/OS-OAUTH1/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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.3.11.1. Request

This table shows the URI parameters for the list roles for an access token request:

Name	Type	Description
{user_id}	Uuid	The ID of the user.
{access_token_id}	Uuid	The ID of the access token.

This operation does not accept a request body.

7.3.12. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.3.12.1. Request

This table shows the URI parameters for the show role details for an access token request:

Name	Type	Description
{user_id}	Uuid	The ID of the user.
{access_token_id}	Uuid	The ID of the access token.
{role_id}	Uuid	The ID of the role.

This operation does not accept a request body.

7.3.12.2. Response

This operation does not return a response body.

7.3.13. 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), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503), Not Found (404)

7.3.13.1. Request

This operation does not accept a request body.

7.3.13.2. Response

This operation does not return a response body.

7.4. 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. A trustee is the user whose trust is being delegated to, 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, any trusts and the related tokens that the user issued with that role are automatically revoked.

For more information, see [Use trusts](#).

Method	URI	Description
POST	/v3/OS-TRUST/trust	Creates a trust.

7.4.1. Create trust

Method	URI	Description
POST	/v3/OS-TRUST/trust	Creates a trust.

Normal response codes: 201

Error response codes: Bad Request (400), Unauthorized (401), Forbidden (403), Method Not Allowed (405), Request Entity Too Large (413), Service Unavailable (503)

7.4.1.1. Request

Example 7.23. 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'"
  }
}
```

7.4.1.2. Response

Example 7.24. 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": {
      "self": "http://localhost:5000/v3/roles-links"
    }
  }
}
```

```
        "next": null,
        "previous": null,
        "self": "http://localhost:5000/v3/OS-TRUST/trusts/
394998fa61f14736b1f0c1f322882949/roles"
    },
    "trustee_user_id": "269348fdd9374b8885da1418e0730af1",
    "trustor_user_id": "3ec3164f750146be97f21559ee4d9c51"
}
}
```

8. 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.
Tokens		
POST	/v2.0/tokens	Authenticates and generates a token.
GET	/v2.0/tenants{?limit,marker}	Lists tenants to which the token has access.

8.1. API versions

Method	URI	Description
GET	/	Lists information about all Identity API versions.
GET	/v2.0	Shows details for the Identity API v2.0.

8.1.1. List versions

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

Normal response codes: 200

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

8.1.1.1. Request

This operation does not accept a request body.

8.1.1.2. Response

Example 8.1. 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+json"
            }
          ]
        }
      ]
    }
  ]
}
```

```
        "type": "application/vnd.openstack.identity-v2.0+json"
    }
],
"status": "stable",
"updated": "2014-04-17T00:00:00Z"
}
]
}
```

8.1.2. Show version details

Method	URI	Description
GET	/v2.0	Shows details for the Identity API v2.0.

Normal response codes: 200 203

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

8.1.2.1. Request

This operation does not accept a request body.

8.1.2.2. Response

Example 8.2. 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"
      }
    ]
  }
}
```

8.2. Tokens

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.

8.2.1. 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: identityFault (400, 500, ...), userDisabled (403), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

8.2.1.1. Request

Example 8.3. Authenticate with user name and password credentials: JSON request

```
{
  "auth": {
    "tenantId": "demo",
    "passwordCredentials": {
      "userId": "demo",
      "password": "secretsecret"
    }
  }
}
```

This table shows the body parameters for the authenticate request:

Name	Type	Description
tenantName	String (Optional)	The tenant name. Both the <code>tenantId</code> and <code>tenantName</code> attributes are optional and mutually exclusive. If you specify both attributes, the server returns the Bad Request (400) response code.
tenantId	UUID (Optional)	The tenant ID. Both the <code>tenantId</code> and <code>tenantName</code> attributes are optional and mutually exclusive. If you specify both attributes, the server returns the Bad Request (400) response code.
passwordCredentials	String (Optional)	A <code>passwordCredentials</code> object. To authenticate, you must provide either a user ID and password or a token.

Name	Type	Description
username	String <i>(Optional)</i>	The user name. Required if you include the passwordCredentials object. Otherwise, you must provide a token.
password	String <i>(Optional)</i>	The password of the user. Required if you include the passwordCredentials object. Otherwise, you must provide a token.
token	String <i>(Optional)</i>	A token object. Required if you do not provide a password credential.
id	String <i>(Optional)</i>	The token ID. This field is required in the token object.

Example 8.4. Authenticate with user name and password credentials: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<auth xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://docs.openstack.org/identity/api/v2.0"
      tenantName="demo">
    <passwordCredentials username="demo" password="secretsecret"/>
</auth>
```

Example 8.5. Authenticate with token: JSON request

```
{
  "auth": {
    "tenantName": "demo",
    "token": {
      "id": "cbc36478b0bd8e67e89469c7749d4127"
    }
  }
}
```

8.2.1.2. Response

Example 8.6. 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": "aaaaaa-bbbb-bcccc-cccc",
      "tenant": {
        "description": null,
        "enabled": true,
        "id": "fc394f2ab2df4114bde39905f800dc57",
        "name": "demo"
      }
    },
    "serviceCatalog": [
      {
        "endpoints": [
          {
            "adminURL": "http://23.253.72.207:8774/v2/
fc394f2ab2df4114bde39905f800dc57",
            "internalURL": "http://23.253.72.207:8774/v2/
fc394f2ab2df4114bde39905f800dc57",
            "publicURL": "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": "97c526db8d7a4c88bbb8d68db1bdedb8",
"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": "93f86dfcbb143a39a33d0c2cd424870",
"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": "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/
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/",
                "id": "16b76b5e5b7d48039a6e4cc3129545f3",
                "publicURL": "http://23.253.72.207:8080/v1/",
                "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",
                "f95c0ab82d6045d9805033ee1fb80d4"
            ]
        },
        "trust": {
            "id": "394998fa61f14736b1f0c1f322882949",
            "trustee_user_id": "269348fdd9374b8885da1418e0730af1",
            "trustor_user_id": "3ec3164f750146be97f21559ee4d9c51",
            "impersonation": false
        }
    }
}
```

This table shows the body parameters for the authenticate response:

Name	Type	Description
access	String <i>(Required)</i>	An access object.
token	String <i>(Required)</i>	A token object.
issued_at	DateTime <i>(Required)</i>	The date and time when the token was issued. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00.
expires	DateTime <i>(Required)</i>	The date and time when the token expires. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. A null value indicates that the token never expires.
id	String <i>(Required)</i>	The authentication token. In the example, the token is my_id.
tenant	String <i>(Required)</i>	A tenant object.
description	String <i>(Required)</i>	The description of the tenant. If not set, this value is null.
enabled	Boolean <i>(Required)</i>	Indicates whether the tenant is enabled or disabled.
id	String <i>(Required)</i>	The tenant ID.
name	String <i>(Required)</i>	The tenant name.
serviceCatalog	String <i>(Required)</i>	A serviceCatalog object.
endpoints	String <i>(Required)</i>	One or more endpoints objects. Each object shows the adminURL, region, internalURL, id, and publicURL for the endpoint.
endpoints_links	String <i>(Required)</i>	Links for the endpoint.
type	String <i>(Required)</i>	Endpoint type.
name	String <i>(Required)</i>	Endpoint name.
user	String	A user object, which shows the username, roles_links, id, roles, and name.

Name	Type	Description
	(Required)	
metadata	String	A metadata object.
	(Required)	
trust	String	A trust object.
	(Optional)	
trustee_user_id	String	The trustee user ID.
	(Optional)	
id	String	The ID of the trust.
	(Optional)	
trustor_user_id	String	The trustor user ID.
	(Optional)	
impersonation	String	The impersonation flag.
	(Optional)	

Example 8.7. Authenticate with user name and password credentials: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<access xmlns="http://docs.openstack.org/identity/api/v2.0">
    <token issued_at="2014-01-30T15:49:11.054709"
        expires="2014-01-31T15:49:11Z"
        id="aaaaaa-bbbb-bcccc-cccc-ccccccccccc">
        <tenant enabled="true" name="demo"
            id="fc394f2ab2df4114bde39905f800dc57"/>
    </token>
    <serviceCatalog>
        <service type="compute" name="nova">
            <endpoints_links/>
            <endpoint
                adminURL="http://23.253.72.207:8774/v2/
fc394f2ab2df4114bde39905f800dc57"
                region="RegionOne"
                publicURL="http://23.253.72.207:8774/v2/
fc394f2ab2df4114bde39905f800dc57"
                internalURL="http://23.253.72.207:8774/v2/
fc394f2ab2df4114bde39905f800dc57"
                id="2dad48f09e2a447a9bf852bcd93548ef">
            </endpoint>
        </service>
        <service type="network" name="neutron">
            <endpoints_links/>
            <endpoint
                adminURL="http://23.253.72.207:9696/"
                region="RegionOne"
                publicURL="http://23.253.72.207:9696/"
                internalURL="http://23.253.72.207:9696/"
                id="97c526db8d7a4c88bbb8d68db1bdcd8">
            </endpoint>
        </service>
        <service type="volumev2" name="cinder">
            <endpoints_links/>
            <endpoint

```

```
    adminURL="http://23.253.72.207:8776/v2/
fc394f2ab2df4114bde39905f800dc57"
        region="RegionOne"
        publicURL="http://23.253.72.207:8776/v2/
fc394f2ab2df4114bde39905f800dc57"
        internalURL="http://23.253.72.207:8776/v2/
fc394f2ab2df4114bde39905f800dc57"
        id="93f86dfcbba143a39a33d0c2cd424870"
    />
</service>
<service type="computev3" name="nova">
    <endpoints_links/>
    <endpoint
        adminURL="http://23.253.72.207:8774/v3"
        region="RegionOne"
        publicURL="http://23.253.72.207:8774/v3"
        internalURL="http://23.253.72.207:8774/v3"
        id="3eb274b12b1d47b2abc536038d87339e"
    />
</service>
<service type="s3" name="s3">
    <endpoints_links/>
    <endpoint adminURL="http://23.253.72.207:3333"
        region="RegionOne"
        publicURL="http://23.253.72.207:3333"
        internalURL="http://23.253.72.207:3333"
        id="957f1e54afc64d33a62099faa5e980a2"
    />
</service>
<service type="image" name="glance">
    <endpoints_links/>
    <endpoint adminURL="http://23.253.72.207:9292"
        region="RegionOne"
        publicURL="http://23.253.72.207:9292"
        internalURL="http://23.253.72.207:9292"
        id="27d5749f36864c7d96bebf84a5ec9767"
    />
</service>
<service type="volume" name="cinder">
    <endpoints_links/>
    <endpoint
        adminURL="http://23.253.72.207:8776/v1/
fc394f2ab2df4114bde39905f800dc57"
        region="RegionOne"
        publicURL="http://23.253.72.207:8776/v1/
fc394f2ab2df4114bde39905f800dc57"
        internalURL="http://23.253.72.207:8776/v1/
fc394f2ab2df4114bde39905f800dc57"
        id="37c83a2157f944f1972e74658aa0b139"
    />
</service>
<service type="ec2" name="ec2">
    <endpoints_links/>
    <endpoint
        adminURL="http://23.253.72.207:8773/services/
Admin"
        region="RegionOne"
        publicURL="http://23.253.72.207:8773/services/
Cloud"
```

```
internalURL="http://23.253.72.207:8773/
services/Cloud"
        id="289b59289d6048e2912b327e5d3240ca"
    />
</service>
<service type="object-store" name="swift">
    <endpoints_links/>
    <endpoint adminURL="http://23.253.72.207:8080"
               region="RegionOne"
               publicURL="http://23.253.72.207:8080/v1/
AUTH_fc394f2ab2df4114bde39905f800dc57"
               internalURL="http://23.253.72.207:8080/v1/
AUTH_fc394f2ab2df4114bde39905f800dc57"
               id="16b76b5e5b7d48039a6e4cc3129545f3"
    />
</service>
<service type="identity" name="keystone">
    <endpoints_links/>
    <endpoint
               adminURL="http://23.253.72.207:35357/v2.0"
               region="RegionOne"
               publicURL="http://23.253.72.207:5000/v2.0"
               internalURL="http://23.253.72.207:5000/v2.0"
               id="26af053673df4ef3a2340c4239e21ea2"
    />
</service>
</serviceCatalog>
<user username="demo" id="9a6590b2ab024747bc2167c4e064d00d"
      name="demo">
    <roles_links/>
    <role name="Member"/>
    <role name="anotherrole"/>
</user>
<metadata is_admin="0">
    <roles>
        <role>7598ac3c634d4c3da4b9126a5f67ca2b</role>
        <role>f95c0ab82d6045d9805033ee1fbc80d4</role>
    </roles>
</metadata>
<trust id="394998fa61f14736b1f0c1f322882949"
       trustee_user_id="269348fdd9374b8885da1418e0730af1"
       trustor_user_id="3ec3164f750146be97f21559ee4d9c51"
       impersonation="false">
</trust>
</access>
```

8.2.2. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

8.2.2.1. Request

This table shows the header parameters for the list tenants request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

Example 8.8. 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.

8.2.2.2. Response

Example 8.9. 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": []
}
```

9. 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.
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.
Users		
POST	/v2.0/users	Creates a user.
GET	/v2.0/users{?name}	Lists all users.
PUT	/v2.0/users/{userId}	Updates a user.
DELETE	/v2.0/users/{userId}	Deletes a user.
GET	/v2.0/users/{user_id}	Gets detailed information about a user by user ID.
GET	/v2.0/users/{user_id}/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}	Gets detailed information about a tenant by name.
GET	/v2.0/tenants/{tenantId}	Gets detailed information about 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.

9.1. Versions

Method	URI	Description
GET	/v2.0	Gets detailed information about a version of the Identity API.

9.1.1. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

9.1.1.1. Request

This operation does not accept a request body.

9.1.1.2. Response

Example 9.1. 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"
      }
    ]
  }
}
```

Example 9.2. 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"
        }
    ]
}
```

9.2. Tokens

Method	URI	Description
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.

9.2.1. 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: identityFault (400, 500, ...), userDisabled (403), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

9.2.1.1. Request

This table shows the header parameters for the authenticate for admin api request:

Name	Type	Description
<code>X-Auth-Token</code>	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 9.3. Authenticate with credentials: JSON request

```
{
  "auth": {
    "tenantName": "admin",
    "passwordCredentials": {
      "username": "admin",
      "password": "secretsecret"
    }
  }
}
```

Example 9.4. Authenticate with token: JSON request

```
{
  "auth": {
```

```
        "tenantName": "demo",
        "token": {
            "id": "cbc36478b0bd8e67e89469c7749d4127"
        }
    }
}
```

9.2.1.2. Response

Example 9.5. 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": "93f86dfcbb143a39a33d0c2cd424870",
        "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": [
            "8341d3603a1d4d5985bff09f10704d4d",
            "2e66d57df76946fdbbe034bc4da6fdec0"
        ]
    },
    "trust": {
        "id": "394998fa61f14736b1f0c1f322882949",
        "trustee_user_id": "269348fdd9374b8885da1418e0730af1",
        "trustor_user_id": "3ec3164f750146be97f21559ee4d9c51",
        "impersonation": false
    }
}
```

Example 9.6. 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": "97c526db8d7a4c88bbb8d68db1bdedb8",
                "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": "93f86dfcbb143a39a33d0c2cd424870",
                "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": "volume",
        "name": "cinder"
    }
]
```

```
        "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": [  
        "8341d3603a1d4d5985bff09f10704d4d",  
        "2e66d57df76946fdbbe034bc4da6fdec0"  
    ]  
},  
"trust": {  
    "id": "394998fa61f14736b1f0c1f322882949",  
    "trustee_user_id": "269348fdd9374b8885da1418e0730af1",  
    "trustor_user_id": "3ec3164f750146be97f21559ee4d9c51",  
    "impersonation": false  
}  
}  
}
```

9.2.2. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

9.2.2.1. Request

This table shows the header parameters for the validate token request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

9.2.2.2. Response

Example 9.7. 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": "345",
          "name": "compute:admin"
        }
      ]
    }
  }
}
```

```
        "id": "234",
        "name": "object-store:admin",
        "tenantId": "1"
    }
],
"roles_links": []
}
}
}
```

Example 9.8. 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": []
    }
  }
}
```

9.2.3. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

9.2.3.1. Request

This table shows the header parameters for the validate token (admin) request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

9.2.4. Delete token

Method	URI	Description
DELETE	/v2.0/tokens/{tokenId}{?belongsTo}	Deletes a token.

Normal response codes: 204

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

9.2.4.1. Request

This table shows the header parameters for the delete token request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

9.3. Users

Method	URI	Description
POST	/v2.0/users	Creates a user.
GET	/v2.0/users{?name}	Lists all users.
PUT	/v2.0/users/{userId}	Updates a user.
DELETE	/v2.0/users/{userId}	Deletes a user.
GET	/v2.0/users/{user_id}	Gets detailed information about a user by user ID.
GET	/v2.0/users/{user_id}/roles{?limit,marker}	Lists global roles for a user. Excludes tenant roles.

9.3.1. Create user

Method	URI	Description
POST	/v2.0/users	Creates a user.

Normal response codes: 201

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

9.3.1.1. Request

This table shows the header parameters for the create user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 9.9. Create user: JSON request

```
{
  "user": {
    "email": "new-user@example.com",
    "password": null,
    "enabled": true,
    "name": "new-user",
    "tenantId": "40429f980fac419bbfec372a5607c154"
  }
}
```

9.3.1.2. Response

Example 9.10. Create user: JSON response

```
{
  "user": {
    "username": "new-user",
    "name": "new-user",
    "id": "71767c619a90479ab21626abf76aa46c",
    "enabled": true,
    "email": "new-user@example.com",
    "tenantId": "40429f980fac419bbfec372a5607c154"
  }
}
```

9.3.2. List users

Method	URI	Description
GET	/v2.0/users{?name}	Lists all users.

To get detailed information about a user by user name, include the `name` query parameter in the request.

Normal response codes: 200203

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

9.3.2.1. Request

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

Name	Type	Description
name	String <i>(Optional)</i>	The user name. Specify the <code>name</code> query parameter as <code>GET /v2.0/users?name={name}</code> .

9.3.2.2. Response

Example 9.11. 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"
        }
    }
```

Example 9.12. 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": "5f58db25afffc44c28d678279981c946f"
        },
        {
            "username": "swiftusertest2",
            "name": "swiftusertest2",
            "enabled": true,
            "email": "test2@example.com",
            "id": "5f58db25afffc44c28d678279981c946f"
        }
    ]
}
```

```
        "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"
    }
}
```

9.3.3. Update user

Method	URI	Description
PUT	/v2.0/users/{userId}	Updates a user.

Normal response codes: 201

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

9.3.3.1. 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 9.13. Update user: JSON request

```
{
  "user": {
    "username": "nova",
    "name": "nova",
    "enabled": false,
    "email": null
  }
}
```

9.3.3.2. Response

Example 9.14. Update user: JSON response

```
{
  "user": {
    "username": "nova",
    "name": "nova",
    "enabled": false,
    "email": null,
    "id": "71767c619a90479ab21626abf76aa46c"
  }
}
```

9.3.4. Delete user

Method	URI	Description
DELETE	/v2.0/users/{userId}	Deletes a user.

Normal response codes: 204

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

9.3.4.1. 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.

9.3.5. Get user information by ID

Method	URI	Description
GET	/v2.0/users/{user_id}	Gets detailed information about a user by user ID.

Normal response codes: 200203

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

9.3.5.1. Request

This table shows the URI parameters for the get user information by id request:

Name	Type	Description
{user_id}	String	The user ID.

This operation does not accept a request body.

9.3.5.2. Response

Example 9.15. Get user information by ID: JSON response

```
{
  "user": {
    "username": "nova",
    "name": "nova",
    "enabled": true,
    "email": null,
    "id": "405ecdef1a434c70bb1e441cd295245d"
  }
}
```

Example 9.16. Get user information by ID: JSON response

```
{
  "user": {
    "username": "nova",
    "name": "nova",
    "enabled": true,
    "email": null,
    "id": "405ecdef1a434c70bb1e441cd295245d"
  }
}
```

9.3.6. List user global roles

Method	URI	Description
GET	/v2.0/users/{user_id}/roles{?limit,marker}	Lists global roles for a user. Excludes tenant roles.

Normal response codes: 200, 203

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

9.3.6.1. Request

This table shows the URI parameters for the list user global roles request:

Name	Type	Description
{user_id}	String	The user ID.

This table shows the query parameters for the list user global roles request:

Name	Type	Description
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

9.3.6.2. Response

Example 9.17. List user global roles: JSON response

```
{
  "roles": [
    {
      "id": "123",
      "name": "compute:admin",
      "description": "Nova Administrator"
    }
  ],
  "roles_links": []
}
```

9.4. Tenants

Method	URI	Description
GET	/v2.0/tenants{?limit,marker}	Lists all tenants.
GET	/v2.0/tenants{?name}	Gets detailed information about a tenant by name.

Method	URI	Description
GET	/v2.0/tenants/{tenantId}	Gets detailed information about 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.

9.4.1. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

9.4.1.1. Request

This table shows the header parameters for the list tenants request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

9.4.1.2. Response

Example 9.18. 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
    }
  ]
}
```

```
        "enabled": true
    }
],
"tenants_links": []
}
```

Example 9.19. 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": []
}
```

9.4.2. Get tenant information by name

Method	URI	Description
GET	/v2.0/tenants{?name}	Gets detailed information about a tenant by name.

Normal response codes: 200203

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

9.4.2.1. Request

This table shows the header parameters for the get tenant information by name request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the query parameters for the get tenant information by name request:

Name	Type	Description
name	String <i>(Required)</i>	The name of the tenant.

This operation does not accept a request body.

9.4.2.2. Response

Example 9.20. Get tenant by name: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

Example 9.21. Get tenant by name: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

9.4.3. Get tenant information by ID

Method	URI	Description
GET	/v2.0/tenants/{tenantId}	Gets detailed information about a tenant by ID.

Normal response codes: 200203

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

9.4.3.1. Request

This table shows the header parameters for the get tenant information by id request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the get tenant information by id request:

Name	Type	Description
{tenantId}	String	The tenant ID.

This operation does not accept a request body.

9.4.3.2. Response

Example 9.22. Get tenant by ID: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

Example 9.23. Get tenant by ID: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

9.4.4. 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: 200, 203

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

9.4.4.1. Request

This table shows the header parameters for the list roles for user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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 <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

9.4.4.2. Response

Example 9.24. List roles for user: JSON response

```
{
  "roles": [
    {
      "id": "123",
      "name": "compute:admin",
      "description": "Nova Administrator"
    }
  ],
  "roles_links": []
}
```

10. 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	/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.
OS-KSCATALOG admin extension		
GET	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints{?limit,marker}	Lists endpoints for a tenant.
POST	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints	Creates endpoint to a tenant.
GET	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints/{endpointId}	Gets endpoint for a tenant.
DELETE	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints/{endpointId}	Deletes an endpoint from a tenant.

Method	URI	Description
GET	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId,limit,marker}	Lists endpoint templates.
POST	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Creates endpoint template.
GET	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Gets endpoint templates.
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}{?type,limit,marker}	Lists credentials by type.
OS-KVALIDATE admin extension		
GET	/v2.0/OS-KVALIDATE/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-KVALIDATE/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-KVALIDATE/token/endpoints{?HP-IDM-serviceId,limit,marker}	Lists endpoints associated with a token.

10.1. OS-KSADM admin extension

Method	URI	Description
GET	/v2.0/users{?limit,marker}	Lists all users.

Method	URI	Description
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.

10.1.1. List users

Method	URI	Description
GET	/v2.0/users{?limit,marker}	Lists all users.

Normal response codes: 200203

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

10.1.1.1. Request

This table shows the header parameters for the list users request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

10.1.1.2. Response

Example 10.1. 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": []
}
```

10.1.2. Create user

Method	URI	Description
POST	/v2.0/users	Creates a user.

Normal response codes: 201

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

10.1.2.1. Request

This table shows the header parameters for the create user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 10.2. Create user: JSON request

```
{
  "user": {
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true,
    "OS-KSADM:password": "secretsecret"
  }
}
```

10.1.2.2. Response

Example 10.3. Create user: JSON response

```
{
  "user": {
    "id": "u1000",
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

10.1.3. Update user

Method	URI	Description
PUT	/v2.0/users/{userId}	Updates a user.

Normal response codes: 200

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

10.1.3.1. Request

This table shows the header parameters for the update user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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 10.4. Update user: JSON request

```
{
  "user": {
    "id": "u1000",
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

10.1.3.2. Response

Example 10.5. Update user: JSON response

```
{
  "user": {
    "id": "u1000",
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

10.1.4. Delete user

Method	URI	Description
DELETE	/v2.0/users/{userId}	Deletes a user.

Normal response codes: 204

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

10.1.4.1. Request

This table shows the header parameters for the delete user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.1.5. Enable user

Method	URI	Description
PUT	/v2.0/users/{userId}/OS-KSADM/enabled	Enables a user.

Normal response codes: 200

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

10.1.5.1. Request

This table shows the header parameters for the enable user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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 10.6. Enable user: JSON request

```
{
  "user": {
    "enabled": true
  }
}
```

10.1.5.2. Response

Example 10.7. Enable user: JSON response

```
{
  "user": {
    "id": "u1000",
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

10.1.6. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.1.6.1. Request

This table shows the header parameters for the list global roles for user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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 <i>(Optional)</i>	The service ID.
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

10.1.6.2. Response

Example 10.8. List global roles for user: JSON response

```
{
  "roles": [
    {
      "id": "8341d3603a1d4d5985bff09f10704d4d",
      "name": "service"
    },
    {
      "id": "2e66d57df76946fdbbe034bc4da6fdec0",
      "name": "user"
    }
  ]
}
```

```
        "name": "admin"
    }
}
```

10.1.7. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

10.1.7.1. Request

This table shows the header parameters for the grant global role to user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.1.8. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.1.8.1. Request

This table shows the header parameters for the delete global role from user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.1.9. Create tenant

Method	URI	Description
POST	/v2.0/tenants	Creates a tenant.

Normal response codes: 201

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415)

10.1.9.1. Request

This table shows the header parameters for the create tenant request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 10.9. Create tenant: JSON request

```
{
  "tenant": {
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

10.1.9.2. Response

Example 10.10. Create tenant: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

10.1.10. Update tenant

Method	URI	Description
POST	/v2.0/tenants/{tenantId}	Updates a tenant.

Normal response codes: 200

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

10.1.10.1. Request

This table shows the header parameters for the update tenant request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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 10.11. Update tenant: JSON request

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

10.1.10.2. Response

Example 10.12. Update tenant: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

10.1.11. Delete tenant

Method	URI	Description
DELETE	/v2.0/tenants/{tenantId}	Deletes a tenant.

Normal response codes: 204

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

10.1.11.1. Request

This table shows the header parameters for the delete tenant request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.1.12. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.1.12.1. Request

This table shows the header parameters for the list users on a tenant request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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 <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

10.1.12.2. Response

Example 10.13. 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
    }
  ]
}
```

```
        "email": "john.smith@example.org",
        "enabled": true
    }
],
"users_links": []
}
```

10.1.13. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

10.1.13.1. Request

This table shows the header parameters for the grant roles to user on tenant request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.1.14. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.1.14.1. Request

This table shows the header parameters for the revoke role from user on tenant request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.1.15. Create role

Method	URI	Description
POST	/v2.0/OS-KSADM	Creates a role.

Normal response codes: 201

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

10.1.15.1. Request

This table shows the header parameters for the create role request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 10.14. Create role: JSON request

```
{
  "role": {
    "id": "123",
    "name": "Guest",
    "description": "Guest Access"
  }
}
```

10.1.15.2. Response

Example 10.15. Create role: JSON response

```
{
  "role": {
    "id": "123",
    "name": "Guest",
    "description": "Guest Access"
  }
}
```

10.1.16. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMediaType (415), itemNotFound (404)

10.1.16.1. Request

This table shows the header parameters for the show role information by name request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.1.16.2. Response

Example 10.16. Show role information by name: JSON response

```
{
    "role": {
        "id": "123",
        "name": "Guest",
        "description": "Guest Access"
    }
}
```

10.1.17. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMediaType (415), itemNotFound (404)

10.1.17.1. Request

This table shows the header parameters for the show role details, by id request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.1.17.2. Response

Example 10.17. Show role details, by ID: JSON response

```
{
  "role": {
    "id": "123",
    "name": "Guest",
    "description": "Guest Access"
  }
}
```

10.1.18. Delete role

Method	URI	Description
DELETE	/v2.0/OS-KSADM/{roleId}	Deletes a role.

Normal response codes: 204

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

10.1.18.1. Request

This table shows the header parameters for the delete role request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.1.19. List roles

Method	URI	Description
GET	/v2.0/OS-KSADM/{?limit,marker}	Lists all roles.

Normal response codes: 200203

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

10.1.19.1. Request

This table shows the header parameters for the list roles request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

10.1.19.2. Response

Example 10.18. List roles: JSON response

```
{
  "roles": [
    {
      "id": "123",
      "name": "compute:admin",
      "description": "Nova Administrator"
    }
  ],
  "roles_links": []
}
```

10.1.20. List services

Method	URI	Description
GET	/v2.0/services{?limit,marker}	Lists all services.

Normal response codes: 200203

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

10.1.20.1. Request

This table shows the header parameters for the list services request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

10.1.20.2. Response

Example 10.19. 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 10.20. 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": []  
}
```

10.1.21. Create service

Method	URI	Description
POST	/v2.0/services	Creates a service.

Normal response codes: 201

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

10.1.21.1. Request

This table shows the header parameters for the create service request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 10.21. Create service: JSON request

```
{
  "OS-KSADM:service": {
    "id": "123",
    "name": "nova",
    "type": "compute",
    "description": "OpenStack Compute Service"
  }
}
```

10.1.21.2. Response

Example 10.22. Create service: JSON response

```
{
  "OS-KSADM:service": {
    "id": "123",
    "name": "nova",
    "type": "compute",
    "description": "OpenStack Compute Service"
  }
}
```

10.1.22. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.1.22.1. Request

This table shows the header parameters for the show service information by name request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.1.22.2. Response

Example 10.23. Show service information by name: JSON response

```
{
    "OS-KSADM:service": {
        "id": "123",
        "name": "nova",
        "type": "compute",
        "description": "OpenStack Compute Service"
    }
}
```

10.1.23. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.1.23.1. Request

This table shows the header parameters for the shows service information by id request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.1.23.2. Response

Example 10.24. Shows service information by ID: JSON response

```
{
    "OS-KSADM:service": {
        "id": "123",
        "name": "nova",
        "type": "compute",
        "description": "OpenStack Compute Service"
    }
}
```

10.1.24. Delete service

Method	URI	Description
DELETE	/v2.0/services/{serviceId}	Deletes a service.

Normal response codes: 204

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

10.1.24.1. Request

This table shows the header parameters for the delete service request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.2. OS-KSCATALOG admin extension

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints{?limit,marker}	Lists endpoints for a tenant.
POST	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints	Creates endpoint to a tenant.
GET	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints/{endpointId}	Gets endpoint for a tenant.
DELETE	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints/{endpointId}	Deletes an endpoint from a tenant.
GET	/v2.0/OS-KSCATALOG/endpointTem-plates{?serviceId,limit,marker}	Lists endpoint templates.
POST	/v2.0/OS-KSCATALOG/endpointTem-plates{?serviceId}	Creates endpoint template.
GET	/v2.0/OS-KSCATALOG/endpointTem-plates/{endpointTemplateId}	Gets endpoint templates.
PUT	/v2.0/OS-KSCATALOG/endpointTem-plates/{endpointTemplateId}	Updates endpoint template.
DELETE	/v2.0/OS-KSCATALOG/endpointTem-plates/{endpointTemplateId}	Deletes an endpoint template.

10.2.1. List endpoints

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints{?limit,marker}	Lists endpoints for a tenant.

Normal response codes: 200 203

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

10.2.1.1. Request

This table shows the header parameters for the list endpoints request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{tenantId}	String	The tenant ID.

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

Name	Type	Description
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

10.2.1.2. Response

Example 10.25. 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",
      "versionId": "1",
    }
  ]
}
```

```
        "versionInfo": "https://compute.north.public.com/v1",
        "versionList": "https://compute.north.public.com/"
    },
    {
        "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",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1",
        "versionList": "https://compute.north.public.com/"
    },
    {
        "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",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1",
        "versionList": "https://compute.north.public.com/"
    },
    {
        "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",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1",
        "versionList": "https://compute.north.public.com/"
    },
    {
        "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",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1",
        "versionList": "https://compute.north.public.com/"
    }
],
"endpoints_links": []
}
```

10.2.2. Create endpoint

Method	URI	Description
POST	/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints	Creates endpoint to a tenant.

Normal response codes: 201

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

10.2.2.1. Request

This table shows the header parameters for the create endpoint request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{tenantId}	String	The tenant ID.

Example 10.26. Create endpoint: JSON request

```
{
  "OS-KSCATALOG:endpointTemplate": {
    "id": 1
  }
}
```

10.2.2.2. Response

Example 10.27. 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",
    "versionId": "1",
    "versionInfo": "https://compute.north.public.com/v1/",
    "versionList": "https://compute.north.public.com/"
  }
}
```

10.2.3. Get endpoint

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints/{endpointId}	Gets endpoint for a tenant.

Normal response codes: 200203

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

10.2.3.1. Request

This table shows the header parameters for the get endpoint request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the get endpoint request:

Name	Type	Description
{tenantId}	String	The tenant ID.
{endpointId}	String	The endpoint ID.

This operation does not accept a request body.

10.2.3.2. Response

Example 10.28. 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",
    "versionId": "1",
    "versionInfo": "https://compute.north.public.com/v1/",
    "versionList": "https://compute.north.public.com/"
  }
}
```

10.2.4. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.2.4.1. Request

This table shows the header parameters for the delete endpoint request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{tenantId}	String	The tenant ID.
{endpointId}	String	The endpoint ID.

This operation does not accept a request body.

10.2.5. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.2.5.1. Request

This table shows the header parameters for the list endpoint templates request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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 <i>(Optional)</i>	The service ID.
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

10.2.5.2. Response

Example 10.29. 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",
      "versionId": "1",
      "versionInfo": "https://compute.north.public.com/v1/",
      "versionList": "https://compute.north.public.com/",
      "enabled": true
    }
  ]
}
```

```
{  
    "id": 2,  
    "region": "South",  
    "type": "compute",  
    "publicURL": "https://compute.south.public.com/v1",  
    "internalURL": "https://compute.south.internal.com/v1",  
    "versionId": "1",  
    "versionInfo": "https://compute.south.public.com/v1/",  
    "versionList": "https://compute.south.public.com/",  
    "enabled": false  
},  
{  
    "id": 3,  
    "region": "North",  
    "global": true,  
    "type": "object-store",  
    "publicURL": "https://object-store.north.public.com/v1.0",  
    "versionId": "1.0",  
    "versionInfo": "https://object-store.north.public.com/v1.0/",  
    "versionList": "https://object-store.north.public.com/",  
    "enabled": true  
},  
{  
    "id": 4,  
    "region": "South",  
    "type": "object-store",  
    "publicURL": "https://object-store.south.public.com/v2",  
    "versionId": "2",  
    "versionInfo": "https://object-store.south.public.com/v2/",  
    "versionList": "https://object-store.south.public.com/",  
    "enabled": true  
},  
{  
    "id": 5,  
    "global": true,  
    "type": "OS-DNS:DNS",  
    "publicURL": "https://dns.public.com/v3.2",  
    "versionId": "1.0",  
    "versionInfo": "https://dns.public.com/v1.0/",  
    "versionList": "https://dns.public.com/",  
    "enabled": true  
}  
],  
"OS-KSCATALOG:endpointsTemplates_links": []  
}
```

10.2.6. Create endpoint template

Method	URI	Description
POST	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Creates endpoint template.

Normal response codes: 201

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

10.2.6.1. Request

This table shows the header parameters for the create endpoint template request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

Example 10.30. 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
    }
}
```

10.2.6.2. Response

Example 10.31. 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",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1/",
        "versionList": "https://compute.north.public.com/",
        "enabled": true
    }
}
```

}

10.2.7. Get endpoint template

Method	URI	Description
GET	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Gets endpoint templates.

Normal response codes: 200203

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

10.2.7.1. Request

This table shows the header parameters for the get endpoint template request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the URI parameters for the get endpoint template request:

Name	Type	Description
{endpointTemplateId}	String	The endpoint template ID.

This operation does not accept a request body.

10.2.7.2. Response

Example 10.32. 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",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1/",
        "versionList": "https://compute.north.public.com/",
        "enabled": true
    }
}
```

10.2.8. Update endpoint template

Method	URI	Description
PUT	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Updates endpoint template.

Normal response codes: 200

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

10.2.8.1. Request

This table shows the header parameters for the update endpoint template request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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 10.33. 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
    }
}
```

10.2.8.2. Response

Example 10.34. 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",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1/",
        "versionList": "https://compute.north.public.com/",
        "enabled": true
    }
}
```

10.2.9. Delete endpoint template.

Method	URI	Description
DELETE	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Deletes an endpoint template.

Normal response codes: 204

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

10.2.9.1. Request

This table shows the header parameters for the delete endpoint template. request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	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.

10.3. OS-KSEC2 admin extension

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.

10.3.1. List credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSADM/credentials{?limit,marker}	Lists credentials.

Normal response codes: 200203

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

10.3.1.1. Request

This table shows the header parameters for the list credentials request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

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

Name	Type	Description
{userId}	String	The user ID.

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

Name	Type	Description
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

10.3.1.2. Response

Example 10.35. List credentials: JSON response

```
{
  "credentials": [
    {
      "passwordCredentials": {
        "username": "test_user",
        "password": "secretsecret"
      }
    },
    {
      "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "accessKey": "AKIAJLZGQH5WV6P6XWQ",
        "secretKey": "wJkDfzqBjC9KuRyOOGHnXGQHdXWYDmJLJLQ",
        "token": "A3114444-4444-4444-4444-444444444444",
        "expiry": "2015-11-17T12:00:00Z"
      }
    }
  ]
}
```

```
        "secret": "secretsecret",
        "signature": "bbb"
    }
},
"credentials_links": []
}
```

10.3.2. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

10.3.2.1. Request

This table shows the header parameters for the grant credential to user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

This table shows the URI parameters for the grant credential to user request:

Name	Type	Description
{userId}	String	The user ID.

Example 10.36. Grant credential to user: JSON request

```
{
    "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "secret": "secretsecret",
        "signature": "bbb"
    }
}
```

10.3.2.2. Response

Example 10.37. Grant credential to user: JSON response

```
{
    "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "secret": "secretsecret",
        "signature": "bbb"
    }
}
```

10.3.3. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.3.3.1. Request

This table shows the header parameters for the get user credentials request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

This table shows the URI parameters for the get user credentials request:

Name	Type	Description
{userId}	String	The user ID.

This operation does not accept a request body.

10.3.3.2. Response

Example 10.38. Get user credentials: JSON response

```
{
  "OS-KSEC2-ec2Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

10.3.4. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

10.3.4.1. Request

This table shows the header parameters for the update user credentials request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

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

Name	Type	Description
{userId}	String	The user ID.

Example 10.39. Update user credentials: JSON request

```
{
    "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "secret": "secretsecret",
        "signature": "bbb"
    }
}
```

10.3.4.2. Response

Example 10.40. Update user credentials: JSON response

```
{
    "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "secret": "secretsecret",
        "signature": "bbb"
    }
}
```

10.3.5. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

10.3.5.1. Request

This table shows the header parameters for the delete user credentials request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

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

Name	Type	Description
{userId}	String	The user ID.

This operation does not accept a request body.

10.3.6. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.3.6.1. Request

This table shows the header parameters for the list credentials by type request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

This table shows the URI parameters for the list credentials by type request:

Name	Type	Description
{userId}	String	The user ID.

This table shows the query parameters for the list credentials by type request:

Name	Type	Description
type	String <i>(Required)</i>	The credential type.
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

10.3.6.2. Response

Example 10.41. List credentials by type: JSON response

```
{
  "credentials": [
    {
      "passwordCredentials": {
        "username": "test_user",
        "password": "secretsecret"
      }
    }
  ]
}
```

```
        }
    ],
    "credentials_links": []
}
```

10.4. 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}{?type,limit,marker}	Lists credentials by type.

10.4.1. List credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSS3/credentials{?limit,marker}	Lists credentials.

Normal response codes: 200203

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

10.4.1.1. Request

This table shows the header parameters for the list credentials request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

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

Name	Type	Description
{userId}	String	The user ID.

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

Name	Type	Description
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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.

10.4.1.2. Response

Example 10.42. List credentials: JSON response

```
{
    "credentials": [
        {
            "passwordCredentials": {
                "username": "test_user",
                "password": "secretsecret"
            }
        },
        {
            "OS-KSS3:s3Credentials": {
                "username": "test_user",
                "password": "secretsecret"
            }
        }
    ]
}
```

```
        "secret": "secretsecret",
        "signature": "bbb"
    }
},
"credentials_links": []
}
```

10.4.2. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

10.4.2.1. Request

This table shows the header parameters for the grant credential to user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

This table shows the URI parameters for the grant credential to user request:

Name	Type	Description
{userId}	String	The user ID.

Example 10.43. Grant credential to user: JSON request

```
{
    "OS-KSS3:s3Credentials": {
        "username": "test_user",
        "secret": "secretsecret",
        "signature": "bbb"
    }
}
```

10.4.2.2. Response

Example 10.44. Grant credential to user: JSON response

```
{
    "OS-KSS3:s3Credentials": {
        "username": "test_user",
        "secret": "secretsecret",
        "signature": "bbb"
    }
}
```

10.4.3. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.4.3.1. Request

This table shows the header parameters for the get user credentials request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

This table shows the URI parameters for the get user credentials request:

Name	Type	Description
{userId}	String	The user ID.

This operation does not accept a request body.

10.4.3.2. Response

Example 10.45. Get user credentials: JSON response

```
{
  "OS-KSS3:s3Credentials": {
    "username": "test_user",
    "secret": "secretsecret",
    "signature": "bbb"
  }
}
```

10.4.4. Update user credentials

Method	URI	Description
POST	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials	Updates credentials.

Normal response codes: 200

Error response codes: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMediaType (415), itemNotFound (404)

10.4.4.1. Request

This table shows the header parameters for the update user credentials request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

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

Name	Type	Description
{userId}	String	The user ID.

Example 10.46. Update user credentials: JSON request

```
{
    "OS-KSS3:s3Credentials": {
        "username": "test_user",
        "secret": "secretsecret",
        "signature": "bbb"
    }
}
```

10.4.4.2. Response

Example 10.47. Update user credentials: JSON response

```
{
    "OS-KSS3:s3Credentials": {
        "username": "test_user",
        "secret": "secretsecret",
        "signature": "bbb"
    }
}
```

10.4.5. 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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

10.4.5.1. Request

This table shows the header parameters for the revoke user credentials request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

This table shows the URI parameters for the revoke user credentials request:

Name	Type	Description
{userId}	String	The user ID.

This operation does not accept a request body.

10.4.6. List credentials by type

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSS3/credentials/s3credentials/{type}{?type,limit,marker}	Lists credentials by type.

Normal response codes: 200203

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

10.4.6.1. Request

This table shows the header parameters for the list credentials by type request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

This table shows the URI parameters for the list credentials by type request:

Name	Type	Description
{userId}	String	The user ID.

This table shows the query parameters for the list credentials by type request:

Name	Type	Description
type	String <i>(Required)</i>	The credential type.
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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 10.48. 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": []
}
```

10.5. OS-KVALIDATE admin extension

Method	URI	Description
GET	/v2.0/OS-KVALIDATE/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-KVALIDATE/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-KVALIDATE/token/endpoints{?HP-IDM-serviceId,limit,marker}	Lists endpoints associated with a token.

10.5.1. Validate token

Method	URI	Description
GET	/v2.0/OS-KVALIDATE/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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.5.1.1. Request

This table shows the header parameters for the validate token request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.
X-Subject-Token	String <i>(Required)</i>	A valid authentication token.

This operation does not accept a request body.

10.5.1.2. Response

Example 10.49. 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": "12345678901234567890123456789012"
        }
      ]
    }
  }
}
```

```
        "id": "234",
        "name": "compute:admin"
    },
    {
        "id": "234",
        "name": "object-store:admin",
        "tenantId": "1"
    }
],
"roles_links": []
}
}
```

10.5.2. Check token

Method	URI	Description
HEAD	/v2.0/OS-KVALIDATE/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: identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

10.5.2.1. Request

This table shows the header parameters for the check token request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.
X-Subject-Token	String <i>(Required)</i>	A valid authentication token.

This operation does not accept a request body.

10.5.3. List endpoints for token

Method	URI	Description
GET	/v2.0/OS-KVALIDATE/token/endpoints{?HP-IDM-serviceId,limit,marker}	Lists endpoints associated with a token.

Normal response codes: 200203

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

10.5.3.1. Request

This table shows the header parameters for the list endpoints for token request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.
X-Subject-Token	String <i>(Required)</i>	A valid authentication token.

This table shows the query parameters for the list endpoints for token request:

Name	Type	Description
HP-IDM-serviceId	String <i>(Optional)</i>	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 <i>(Optional)</i>	Requests a page size of returned items from the query. 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 <i>(Optional)</i>	Specifies 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.

10.5.3.2. Response

Example 10.50. 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",
      "status": "OK"
    }
  ]
}
```

```
        "adminURL": "https://compute.north.internal.com/v1",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1/",
        "versionList": "https://compute.north.public.com/"
    },
    {
        "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",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1/",
        "versionList": "https://compute.north.public.com/"
    },
    {
        "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",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1/",
        "versionList": "https://compute.north.public.com/"
    },
    {
        "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",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1/",
        "versionList": "https://compute.north.public.com/"
    },
    {
        "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",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1/",
        "versionList": "https://compute.north.public.com/"
    }
],
"endpoints_links": []
}
```

11. Image service API v2 (CURRENT)

Image service API v2.0, API v2.1, and API v2.2.

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.
Images		
POST	/v2/images	(Since Image API v2.0) Creates a virtual machine (VM) image.
GET	/v2/images{?limit,marker,name,visibility,member_status,owner,status,size_min,size_max,sort_key,sort_dir,sort,tag}	(Since Image API v2.0) Lists public virtual machine (VM) images.
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.
Image data		
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		
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		
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	(Since Image API v2.1) Lists the tenants with whom this image has been shared.
GET	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.2) Shows image member details.
DELETE	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.1) Deletes a tenant ID from the member list of an image.
PUT	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.1) Sets the status for an image member.
Image schemas		
GET	/v2/schemas/images	Gets a json-schema document that represents an images entity. (Since Images v2.0.)
GET	/v2/schemas/image	Gets a json-schema document that represents an image entity. (Since Images v2.0.)
GET	/v2/schemas/members	Gets a json-schema document that represents an image members entity. (Since Images v2.1.)
GET	/v2/schemas/member	Gets a json-schema document that represents an image member entity. (Since Images v2.1.)

Method	URI	Description
Metadata definition resource types (since API v2.0)		
GET	/v2/metadefs/resource_types	Lists resource types.
POST	/v2/metadefs/namespaces/{namespace}/resource_types	Creates a resource type association in a namespace.
GET	/v2/metadefs/namespaces/{namespace}/resource_types	Lists resource type associations in a namespace.
DELETE	/v2/metadefs/namespaces/{namespace}/resource_types/{name}	Removes a resource type association in a namespace.
Metadata definition namespaces (since API v2.0)		
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}	Gets details for a namespace.
PUT	/v2/metadefs/namespaces/{namespace}	Updates a namespace.
DELETE	/v2/metadefs/namespaces/{namespace}	Deletes a namespace and its properties, objects, and any resource type associations.
Metadata definition properties (since API v2.0)		
POST	/v2/metadefs/namespaces/{namespace}/properties	Creates a property definition within a namespace.
GET	/v2/metadefs/namespaces/{namespace}/properties	Lists property definitions within a namespace.
GET	/v2/metadefs/namespaces/{namespace}/properties/{property_name}{?resource_type}	Gets the definition for a property.
PUT	/v2/metadefs/namespaces/{namespace}/properties/{property_name}	Updates a property definition.
DELETE	/v2/metadefs/namespaces/{namespace}/properties/{property_name}	Removes a property definition in a namespace.
Metadata definition objects (since API v2.0)		
POST	/v2/metadefs/namespaces/{namespace}/objects	Creates an object definition in a namespace.
GET	/v2/metadefs/namespaces/{namespace}/objects{?visibility,resource_types,sort_key,sort_dir}	Lists object definitions within a namespace.
GET	/v2/metadefs/namespaces/{namespace}/objects/{object_name}	Gets the definition for an object.
PUT	/v2/metadefs/namespaces/{namespace}/objects/{object_name}	Updates an object definition in a namespace.
DELETE	/v2/metadefs/namespaces/{namespace}/objects/{object_name}	Deletes an object definition within a namespace.
Metadata definition tags (since API v2.0)		
POST	/v2/metadefs/namespaces/{namespace}/tags	Creates one or more tag definitions in a namespace.
GET	/v2/metadefs/namespaces/{namespace}/tags{?limit,marker,sort_key,sort_dir}	Lists the tag definitions within a namespace.
DELETE	/v2/metadefs/namespaces/{namespace}/tags	Deletes all tag definitions within a namespace.

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace}/tags/{name}	Adds a tag to the list of namespace tag definitions.
GET	/v2/metadefs/namespaces/{namespace}/tags/{name}	Gets a definition for a tag.
PUT	/v2/metadefs/namespaces/{namespace}/tags/{name}	Renames a tag definition.
DELETE	/v2/metadefs/namespaces/{namespace}/tags/{name}	Deletes a tag definition within a namespace.
Metadata definition schemas		
GET	/v2/schemas/metadefs/namespace	Gets a JSON schema document that represents a metadata definition namespace entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/namespaces	Gets a JSON schema document that represents a metadata definition namespaces entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/object	Gets a JSON schema document that represents a metadata definition object entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/objects	Gets a JSON schema document that represents a metadata definition objects entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/property	Gets a JSON schema document that represents a metadata definition property entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/properties	Gets a JSON schema document that represents a metadata definition properties entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/tag	Gets a JSON schema document that represents a metadata definition tag entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/tags	Gets a JSON schema document that represents a metadata definition tags entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/resource_type	Gets a JSON schema document that represents a metadata definition namespace resource type association entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/resource_types	Gets a JSON schema document that represents a metadata definition namespace resource type associations entity. (Since API v2.1.)
Tasks (since API v2.2)		
POST	/v2/tasks	Creates a task.
GET	/v2/tasks{?type,status,sort_key,sort_dir}	Lists tasks.
GET	/v2/tasks/{task_id}	Shows details for a task.

11.1. API versions

Method	URI	Description
GET	/	Lists information about all Image service API versions.

11.1.1. List API versions

Method	URI	Description
GET	/	Lists information about all Image service API versions.

Normal response codes: 200, 300

11.1.1.1. Request

This operation does not accept a request body.

11.1.1.2. Response

Example 11.1. List API versions: JSON response

```
{
    "versions": [
        {
            "status": "CURRENT",
            "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"
            }
        ]
    }
}

```

11.2. 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 11.1. Image status

Status	Description
queued	The Image service reserved an image ID for the image in the registry but has not uploaded 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 error occurred during the upload of image data.
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
POST	/v2/images	(Since Image API v2.0) Creates a virtual machine (VM) image.
GET	/v2/images{?limit,marker,name,visibility,member_status,owner,status,size_min,size_max,sort_key,sort_dir,sort,tag}	(Since Image API v2.0) Lists public virtual machine (VM) images.
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.

11.2.1. Create image

Method	URI	Description
POST	/v2/images	(Since Image API v2.0) Creates a virtual machine (VM) image.

Created with a `Location` header that contains the newly-created URI for the image. Response body represents the created image entity.

Synchronous Postconditions

- With correct permissions, you can see the image status as `queued` through API calls.
- With correct permissions, you can see the image status as `active` through API calls if you created the image with `Location` header.

Normal response codes: 201

11.2.1.1. Request

Example 11.2. Create image: JSON request

```
{
  "name": "Ubuntu 12.10",
  "tags": [
    "ubuntu",
    "quantal"
  ],
  "container_format": "bare",
  "disk_format": "qcow2",
  "visibility": "private",
  "min_disk": 0,
  "protected": false,
  "min_ram": 0
}
```

11.2.1.2. Response

Example 11.3. Create image: JSON response

```
{
  "status": "queued",
  "name": "Ubuntu 12.10",
  "tags": [
    "ubuntu",
    "quantal"
  ],
  "container_format": "bare",
  "created_at": "2014-11-11T20:47:55Z",
  "disk_format": "qcow2",
  "updated_at": "2014-11-11T20:47:55Z",
  "visibility": "private",
  "self": "/v2/images/b2173dd3-7ad6-4362-baa6-a68bce3565ca",
  "min_disk": 0,
  "protected": false,
```

```
    "id": "b2173dd3-7ad6-4362-baa6-a68bce3565ca",
    "file": "/v2/images/b2173dd3-7ad6-4362-baa6-a68bce3565ca/file",
    "owner": "b4eedccc6fb74fa8a7ad6b08382b852b",
    "min_ram": 0,
    "schema": "/v2/schemas/image",
    "size": null,
    "checksum": null,
    "virtual_size": null
}
```

11.2.2. 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}	(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 several types of query parameters that let you filter the results of the returned collection.

A client can provide direct comparison filters using most image attributes, such as `name=Ubuntu`, `visibility=public`, and so on. A client cannot filter on tags or anything defined as a link in the json-schema, such as `self`, `file`, or `schema`.

You can use the `size_min` and `size_max` query parameters to perform greater-than and less-than filtering of images based on their `size` attribute. The `size` is measured in bytes and refers to the size of an image when it is stored on disk.

For example, sending a `size_min` filter of 1048576 and `size_max` of 4194304 filters the container to include only images that are between 1 and 4 MB.

You can see a list of VM images which status is in `active`, `queued`, or `saving`.

You can use query parameters to sort the results of this operation.

- `sort_key`. Sorts by the requested image attribute. Sorts in the natural sorting direction of the image attribute that is provided as the `sort_key`.
- `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 specify the `sort_key` and `sort_dir` query parameters to sort the results:

```
GET /v2/images?sort_key=name&sort_dir=asc&sort_key=status&sort_dir=desc
```

To specify the `sort` query parameter to sort the results:

```
GET /v2/images?sort=name:asc,status:desc
```

Normal response codes: 200

11.2.2.1. Request

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

Name	Type	Description
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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 <i>(Optional)</i>	Filter parameter. Shows only images with this name. A valid value is the name of the image as a string.
visibility	String <i>(Optional)</i>	Filter parameter. Shows only images with this image visibility value or values. Valid values are <code>public</code> , <code>private</code> , and <code>shared</code> . If you omit this parameter, the response shows <code>public</code> , <code>private</code> , and <code>shared</code> images with accepted member status.
member_status	String <i>(Optional)</i>	Filter parameter. Shows only images with this 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 <i>(Optional)</i>	Filter parameter. Shows only images that are shared with this owner. A valid value is a tenant ID.
status	Int <i>(Optional)</i>	Filter parameter. Shows only images with this 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 <i>(Optional)</i>	Filter parameter. Shows only images with this minimum image size. A valid value is the minimum size of the image, in bytes.
size_max	String <i>(Optional)</i>	Filter parameter. Shows only images with this maximum image size. A valid value is the maximum size of the image, in bytes.
sort_key	String <i>(Optional)</i>	Sorts by the requested image attribute. Accepted values are <code>name</code> , <code>status</code> , <code>container_format</code> , <code>disk_format</code> , <code>size</code> , <code>id</code> , <code>created_at</code> , and <code>updated_at</code> . Default is <code>created_at</code> . The API uses the natural sorting direction of the image attribute that is provided as the <code>sort_key</code> .
sort_dir	String <i>(Optional)</i>	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, the default is <code>desc</code> .

Name	Type	Description
sort	String <i>(Optional)</i>	Sorts by one or more attribute and sort direction combinations. You can also set multiple sort keys and directions. Default direction is desc. For example: <code>GET /v2/images?sort=name:asc,status:desc</code>
tag	String <i>(Optional)</i>	Image tag. For example, ?tag="cirros".

This operation does not accept a request body.

11.2.2.2. Response

Example 11.4. 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"
}
```

11.2.3. 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)

11.2.3.1. Request

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

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

11.2.3.2. Response

Example 11.5. 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
}
```

11.2.4. 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 11.2. Image update actions

Target location	Update action
An array index	A new value is inserted into the array at the index.
An object member that does not exist	A new member is added 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 to add to an object. The value is added to that object at that location. If the member already exists, it is replaced by the value.
- An element to add to the array. The value is added to the array at the location. Any element that is at or above the index is shifted 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 value is appended 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

11.2.4.1. Request

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

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.

Example 11.6. Update image: JSON request

```
[
  {
    "op": "replace",
    "path": "/name",
    "value": "Fedora 17"
  },
  {
    "op": "replace",
    "path": "/tags",
    "value": [
      "fedora",
      "beefy"
    ]
  }
]
```

11.2.4.2. Response

Example 11.7. 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
}
```

11.2.5. 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 all status except deleted.
- You must first set the `protected` attribute to false (boolean) and then perform the delete.

Synchronous Postconditions

- The response is empty and returns the HTTP 204 status code.
- The image is deleted in images index.
- The binary image data managed by OpenStack Image service is deleted from the storage node if the deleted image stores image data in the node.

Troubleshooting

- The response returns the HTTP 403 status 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)

11.2.5.1. Request

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

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

11.2.6. Reactivate image

Method	URI	Description
POST	/v2/images/{image_id}/actions/reactivate	(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

11.2.6.1. Request

This table shows the URI parameters for the reactivate image request:

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

11.2.6.2. Response

Example 11.8. Reactivate image: 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
}
```

11.2.7. 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 403 Forbidden error is returned. Also, image locations are not visible for deactivated images unless the user is an administrative user.

The deactivate operation returns an error if the image status is not active or deactivated.

Preconditions

- The image must exist.

Normal response codes: 204

11.2.7.1. Request

This table shows the URI parameters for the deactivate image request:

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

11.2.7.2. Response

Example 11.9. Deactivate image: JSON response

```
{
    "status": "deactivated",
    "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
}
```

11.3. Image data

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.

11.3.1. 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, your request might lack 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

11.3.1.1. Request

This table shows the URI parameters for the upload binary image data request:

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

11.3.2. 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. Clients are encouraged to use this checksum to verify the integrity of the image data that they receive.

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 status code.
- If no image data exists, the call returns the HTTP 204 status code.

Normal response codes:

200, 204

Error response codes:

forbidden (403)

11.3.2.1. 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	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

11.4. Image 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.

11.4.1. 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

11.4.1.1. Request

This table shows the URI parameters for the add image tag request:

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.
{tag}	String	Image tag.

This operation does not accept a request body.

11.4.2. 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

11.4.2.1. Request

This table shows the URI parameters for the delete image tag request:

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.
{tag}	String	Image tag.

This operation does not accept a request body.

11.5. Members

Method	URI	Description
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	(Since Image API v2.1) Lists the tenants with whom this image has been shared.
GET	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.2) Shows image member details.
DELETE	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.1) Deletes a tenant ID from the member list of an image.
PUT	/v2/images/{image_id}/members/{member_id}	(Since Image API v2.1) Sets the status for an image member.

11.5.1. 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.

Preconditions

- The images must exist.
- You can only add a new member to an image which 'visibility' attribute is private.
- 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)

11.5.1.1. Request

This table shows the URI parameters for the create image member request:

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.

Example 11.10. Create image member: JSON request

```
{
    "member": "8989447062e04a818baf9e073fd04fa7"
}
```

11.5.1.2. Response

Example 11.11. 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"
}
```

11.5.2. List image members

Method	URI	Description
GET	/v2/images/{image_id}/members	(Since Image API v2.1) Lists the tenants with whom this image has been shared.

If a user with whom this image is shared makes this call, the member list contains only information for that user.

If a user with whom this image has not been shared makes this call, the call returns the HTTP 404 status code.

Preconditions

- The image must exist.
- You must be the owner or a member of the image.

Normal response codes: 200

Error response codes: 404

11.5.2.1. Request

This table shows the URI parameters for the list image members request:

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

11.5.2.2. Response

Example 11.12. 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"  
    }
```

11.5.3. 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

11.5.3.1. Request

This table shows the URI parameters for the show image member details request:

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.
{member_id}	String	Image member ID. For example, the tenant ID of the user with whom the image is being shared.

This operation does not accept a request body.

11.5.3.2. Response

Example 11.13. 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"
}
```

11.5.4. 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 member is removed from the image members.

Troubleshooting

- Even if you have correct permissions, if you are not the owner of the image, the request returns the HTTP 403 error code. 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)

11.5.4.1. Request

This table shows the URI parameters for the delete image member request:

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.
{member_id}	String	Image member ID. For example, the tenant ID of the user with whom the image is being shared.

This operation does not accept a request body.

11.5.5. 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 see the updated member status of the image through API calls.

Normal response codes: 200

11.5.5.1. Request

This table shows the URI parameters for the update image member request:

Name	Type	Description
{image_id}	Uuid	Image ID stored through the image API. Typically a UUID.
{member_id}	String	Image member ID. For example, the tenant ID of the user with whom the image is being shared.

Example 11.14. Update image member: JSON request

```
{
    "status": "accepted"
}
```

11.5.5.2. Response

Example 11.15. 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"
}
```

11.6. Image schemas

Gets a JSON-schema document that represents an images or image entity.

Method	URI	Description
GET	/v2/schemas/images	Gets a json-schema document that represents an images entity. (Since Images v2.0.)
GET	/v2/schemas/image	Gets a json-schema document that represents an image entity. (Since Images v2.0.)
GET	/v2/schemas/members	Gets a json-schema document that represents an image members entity. (Since Images v2.1.)
GET	/v2/schemas/member	Gets a json-schema document that represents an image member entity. (Since Images v2.1.)

11.6.1. Get images schema

Method	URI	Description
GET	/v2/schemas/images	Gets a json-schema document that represents an images entity. (Since Images v2.0.)

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

11.6.1.1. Request

This operation does not accept a request body.

11.6.1.2. Response

Example 11.16. Get 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"  
                    }  
                ]  
            }  
        }  
    }  
}
```

```
        "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}$",
        "type": "string"
    },
    "instance_uuid": {
        "description": "ID of instance used to create this
image.",
        "is_base": false,
        "type": "string"
    },
    "kernel_id": {
        "description": "ID of image stored in Glance that
should be used as the kernel when booting an AMI-style image.",
        "is_base": false,
        "pattern": "^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}$",
        "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"
}
}
}
```

11.6.2. Get image schema

Method	URI	Description
GET	/v2/schemas/image	Gets a json-schema document that represents an image entity. (Since Images v2.0.)

The following schema is solely an example. Consider only the response to the API call as authoritative.

Normal response codes: 200

11.6.2.1. Request

This operation does not accept a request body.

11.6.2.2. Response

Example 11.17. Get 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": [
                "ami",
                "aki",
                "ari",
                "raw",
                "qcow2",
                "vhd",
                "vmdk",
                "iso",
                "vdi",
                "vpc"
            ]
        }
    }
}
```

```
        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]{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"  
        ]  
    }  
}
```

```
        "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"
    }
}
```

11.6.3. Get image members schema

Method	URI	Description
GET	/v2/schemas/members	Gets a json-schema document that represents an image members entity. (Since Images v2.1.)

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

11.6.3.1. Request

This operation does not accept a request body.

11.6.3.2. Response

Example 11.18. Get 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]{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": [
                            "active"
                        ]
                    }
                }
            }
        }
    }
}
```

```
        "pending",
        "accepted",
        "rejected"
    ],
    "type": "string"
},
"updated_at": {
    "description": "Date and time of last modification of
image member",
    "type": "string"
}
},
"schema": {
    "type": "string"
}
}
```

11.6.4. Get image member schema

Method	URI	Description
GET	/v2/schemas/member	Gets a json-schema document that represents an image member entity. (Since Images v2.1.)

The following schema is solely an example. Consider only the response to the API call as authoritative.

Normal response codes: 200

11.6.4.1. Request

This operation does not accept a request body.

11.6.4.2. Response

Example 11.19. Get 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"
        }
    }
}
```

11.7. Metadata definition resource types (since API v2.0)

Lists resource types. Also, creates, lists, and removes resource type associations in a namespace.

Method	URI	Description
GET	/v2/metadefs/resource_types	Lists resource types.
POST	/v2/metadefs/namespaces/{namespace}/resource_types	Creates a resource type association in a namespace.
GET	/v2/metadefs/namespaces/{namespace}/resource_types	Lists resource type associations in a namespace.
DELETE	/v2/metadefs/namespaces/{namespace}/resource_types/{name}	Removes a resource type association in a namespace.

11.7.1. List resource types

Method	URI	Description
GET	/v2/metadefs/resource_types	Lists resource types.

Lists all possible resource types. You can assign metadata definition namespaces to these resource types. See the metadata definition resource types section.

Normal response codes: 200

11.7.1.1. Request

This operation does not accept a request body.

11.7.1.2. Response

Example 11.20. 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"
        }
    ]
}
```

11.7.2. Create resource type association

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace}/resource_types	Creates a resource type association in a namespace.

Normal response codes: 201

11.7.2.1. Request

This table shows the URI parameters for the create resource type association request:

Name	Type	Description
{namespace}	String	Unique namespace.

Example 11.21. Create resource type association: JSON request

```
{
    "name": "OS::Cinder::Volume",
    "prefix": "hw_",
    "properties_target": "image"
}
```

11.7.2.2. Response

Example 11.22. 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"
}
```

11.7.3. List resource type associations

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace}/resource_types	Lists resource type associations in a namespace.

The response body lists resource type association entities.

Normal response codes: 200

11.7.3.1. Request

This table shows the URI parameters for the list resource type associations request:

Name	Type	Description
{namespace}	String	Unique namespace.

This operation does not accept a request body.

11.7.3.2. Response

Example 11.23. List resource type associations: JSON response

```
{
    "resource_type_associations": [
        {
            "created_at": "2014-09-19T16:13:33Z",
            "name": "OS::Glance::Image",
            "prefix": "hw_",
            "updated_at": "2014-09-19T16:13:33Z"
        },
        {
            "created_at": "2014-09-19T16:09:13Z",
            "name": "OS::Cinder::Volume",
            "prefix": "hw_",
            "properties_target": "image",
            "updated_at": "2014-09-19T16:09:13Z"
        }
    ]
}
```

11.7.4. Remove resource type association

Method	URI	Description
DELETE	/v2/metadefs/namespaces/{namespace}/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: 403

11.7.4.1. Request

This table shows the URI parameters for the remove resource type association request:

Name	Type	Description
{namespace}	String	Unique namespace.
{name}	String	Name of the resource type.

This operation does not accept a request body.

11.8. Metadata definition namespaces (since API v2.0)

Creates, lists, gets 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}	Gets details for a namespace.
PUT	/v2/metadefs/namespaces/{namespace}	Updates a namespace.
DELETE	/v2/metadefs/namespaces/{namespace}	Deletes a namespace and its properties, objects, and any resource type associations.

11.8.1. Create namespace

Method	URI	Description
POST	/v2/metadefs/namespaces	Creates a namespace.

The namespace is created with a `Location` header that contains the newly-created URI for the namespace.

Normal response codes: 201

11.8.1.1. Request

Example 11.24. 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"
}
```

11.8.1.2. Response

Example 11.25. 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"  
}
```

11.8.2. 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 let you filter the results in the returned collection.

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

11.8.2.1. Request

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

Name	Type	Description
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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 <i>(Optional)</i>	Filter parameter. Namespace visibility. Valid values are <code>public</code> and <code>private</code> . Default is <code>public</code> .
resource_types	Int <i>(Optional)</i>	Filter parameter. Specify this value as comma-separated list. For example, send the <code>resource_types</code> filter of <code>OS::Glance::Image,OS::Nova::Flavor</code> to filter the namespaces by those resource types.
sort_key	String <i>(Optional)</i>	Sort key. Use attributes like <code>namespace</code> for sorting. Default is <code>created_at</code> .
sort_dir	String <i>(Optional)</i>	Sort direction. Valid values are <code>asc</code> (ascending) and <code>desc</code> (descending). Default is <code>desc</code> .

This operation does not accept a request body.

11.8.2.2. Response

Example 11.26. 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::ComputePool",
                    "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"
        }
    ]
}
```

```
"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"
```

11.8.3. Get namespaces details

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace}	Gets details for a namespace.

The response body shows a single namespace entity with all details including properties and objects.

Normal response codes: 200

11.8.3.1. Request

This table shows the URI parameters for the get namespaces details request:

Name	Type	Description
{namespace}	String	Unique namespace.

This operation does not accept a request body.

11.8.3.2. Response

Example 11.27. 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": "Specifies the QEMU guest agent to use for the machine. This is typically used for ARM images to enable features like KVM acceleration. Valid values are 'none' or 'qemu'. The default value is 'qemu'. Note that this setting only applies to ARM images; for other architectures, the guest agent is determined by the machine type or image configuration."}
    }
}
```

```
        "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, rtl18139, and virtio. VMware: e1000, e1000e, VirtualE1000, VirtualE1000e, VirtualPCNet32, VirtualSriovEthernetCard, and VirtualVmxnet. Xen: e1000, netfront, ne2k_pci, pcnet, and rtl18139.",
        "enum": [
            "e1000",
            "e1000e",
            "VirtualE1000",
            "VirtualE1000e",
            "VirtualPCNet32",
            "VirtualSriovEthernetCard",
            "VirtualVmxnet",
            "ne2k_pci",
            "pcnet",
            "rtl18139",
            "netfront"
        ],
        "title": "Virtual Network Interface Card Model",
        "type": "string"
    }
}
```

```
        "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"
}
```

11.8.4. Update namespace

Method	URI	Description
PUT	/v2/metadefs/namespaces/{namespace}	Updates a namespace.

Normal response codes: 200

11.8.4.1. Request

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

Name	Type	Description
{namespace}	String	Unique namespace.
{namespace}	Uuid	Unique namespace.

Example 11.28. 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"
}
```

11.8.4.2. Response

Example 11.29. 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 harware 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"
}
```

11.8.5. Delete namespace

Method	URI	Description
DELETE	/v2/metadefs/namespaces/{namespace}	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 status 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.

If the operation succeeds, the response returns the HTTP 204 status 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 error code.

Normal response codes: 204

Error response codes: 403

11.8.5.1. Request

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

Name	Type	Description
{namespace}	String	Unique namespace.

This operation does not accept a request body.

11.9. Metadata definition properties (since API v2.0)

Creates, lists, gets details for, updates, and deletes metadata definition properties.

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace}/properties	Creates a property definition within a namespace.
GET	/v2/metadefs/namespaces/{namespace}/properties	Lists property definitions within a namespace.
GET	/v2/metadefs/namespaces/{namespace}/properties/{property_name} {?resource_type}	Gets the definition for a property.
PUT	/v2/metadefs/namespaces/{namespace}/properties/{property_name}	Updates a property definition.
DELETE	/v2/metadefs/namespaces/{namespace}/properties/{property_name}	Removes a property definition in a namespace.

11.9.1. Create property

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace}/properties	Creates a property definition within a namespace.

The schema is a subset of JSON property definition schema.

Normal response codes: 201

11.9.1.1. Request

Example 11.30. 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"
}
```

11.9.1.2. Response

Example 11.31. 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"
}
```

}

11.9.2. List properties

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace}/properties	Lists property definitions within a namespace.

Returns a list of property definitions in a namespace.

Normal response codes: 200

11.9.2.1. Request

This operation does not accept a request body.

11.9.2.2. Response

Example 11.32. List properties: 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": {
            "description": "Specifies the type of random number generator to use. Valid values are trng, i64, and none. If none is selected, no random number generator will be used. If trng is selected, Compute will attempt to use the Intel(R) Software Guard Extensions (SGX) TRNG. If i64 is selected, Compute will attempt to use the Intel(R) Software Guard Extensions (SGX) I64 RNG. If none of these are available, Compute will fall back to using the Linux kernel's built-in random number generator (RNG).",
            "enum": [
                "trng",
                "i64",
                "none"
            ],
            "title": "Hardware Random Number Generator Model",
            "type": "string"
        }
    }
}
```

```
        "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, rtl18139, and virtio. VMware: e1000, e1000e,
VirtualE1000, VirtualE1000e, VirtualPCNet32, VirtualSriovEthernetCard, and
VirtualVmxnet. Xen: e1000, netfront, ne2k_pci, pcnet, and rtl18139.",
        "enum": [
            "e1000",
            "ne2k_pci",
            "pcnet",
            "rtl18139",
            "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"
    }
}
```

11.9.3. Get property definition

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace}/properties/{property_name} {?resource_type}	Gets the definition for a property.

If you include the `resource_type` query parameter, the prefix of the included resource type is removed from the property name before the query is submitted. This enables you to look for a property name that starts with a prefix from an associated resource type.

Response body shows a single property entity.

Normal response codes: 200

11.9.3.1. Request

This table shows the query parameters for the get property definition request:

Name	Type	Description
resource_type	Int <i>(Optional)</i>	Filter parameter. Shows only property names that start with a prefix from an associated resource type. The prefix of the included resource type is removed from the property name in the response.

This operation does not accept a request body.

11.9.3.2. Response

Example 11.33. Get 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"
}
```

11.9.4. Update property definition

Method	URI	Description
PUT	/v2/metadefs/namespaces/{namespace}/properties/{property_name}	Updates a property definition.

Normal response codes: 200

11.9.4.1. Request

Example 11.34. 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"
}
```

11.9.4.2. Response

Example 11.35. 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"
}
```

11.9.5. Remove property definition

Method	URI	Description
DELETE	/v2/metadefs/namespaces/{namespace}/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: 403

11.9.5.1. Request

This operation does not accept a request body.

11.10. Metadata definition objects (since API v2.0)

Creates, lists, gets details for, updates, and deletes metadata definition objects.

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace}/objects	Creates an object definition in a namespace.
GET	/v2/metadefs/namespaces/{namespace}/objects{?visibility, resource_types, sort_key, sort_dir}	Lists object definitions within a namespace.
GET	/v2/metadefs/namespaces/{namespace}/objects/{object_name}	Gets the definition for an object.
PUT	/v2/metadefs/namespaces/{namespace}/objects/{object_name}	Updates an object definition in a namespace.
DELETE	/v2/metadefs/namespaces/{namespace}/objects/{object_name}	Deletes an object definition within a namespace.

11.10.1. Create object

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace}/objects	Creates an object definition in a namespace.

Normal response codes: 201

11.10.1.1. Request

Example 11.36. 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": []
}
```

11.10.1.2. Response

Example 11.37. 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"
}
```

11.10.2. List objects

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace}/objects{?visibility,resource_types,sort_key,sort_dir}	Lists object definitions within 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.

The list operation accepts `resource_types` and `visibility` as query parameters that let you filter the results of the returned collection.

For example, sending a `resource_types` filter of `OS::Glance::Image,OS::Nova::Flavor` filters the namespaces to include only namespaces that are associated to 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

11.10.2.1. Request

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

Name	Type	Description
<code>visibility</code>	String <i>(Optional)</i>	Filter parameter. Shows only namespaces with this visibility value or values. Valid values are <code>public</code> and <code>private</code> . If you omit this parameter, the response shows <code>public</code> and <code>private</code> namespaces.
<code>resource_types</code>	Int <i>(Optional)</i>	Filter parameter. Shows only namespaces with this resource type or types. Specify multiple values as a comma-separated list. For example, set the <code>resource_types</code> filter to <code>OS::Glance::Image,OS::Nova::Flavor</code> to include only namespaces that are associated with this resource types.
<code>sort_key</code>	String <i>(Optional)</i>	Sort key. A valid value is an attribute, such as <code>namespace</code> , for sorting. Default is <code>created_at</code> .
<code>sort_dir</code>	String <i>(Optional)</i>	Sort direction. A valid value is <code>asc</code> (ascending) or <code>desc</code> (descending). Default is <code>desc</code> .

This operation does not accept a request body.

11.10.2.2. Response

Example 11.38. List objects: 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 and 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"
}
```

11.10.3. Get object definition

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace}/objects/{object_name}	Gets the definition for an object.

Response body is a single object entity.

Normal response codes: 200

11.10.3.1. Request

This table shows the URI parameters for the get object definition request:

Name	Type	Description
{namespace}	Uuid	Unique namespace.
{object_name}	String	The name of the object.

This operation does not accept a request body.

11.10.3.2. Response

Example 11.39. Get 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"
}
```

11.10.4. Update object definition

Method	URI	Description
PUT	/v2/metadefs/namespaces/{namespace}/objects/{object_name}	Updates an object definition in a namespace.

Normal response codes: 200

11.10.4.1. Request

This table shows the URI parameters for the update object definition request:

Name	Type	Description
{namespace}	Uuid	Unique namespace.
{object_name}	String	The name of the object.

Example 11.40. 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": []
}
```

11.10.4.2. Response

Example 11.41. 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/metadefs/object",  
    "self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/CPU_Limits",  
    "updated_at": "2014-09-19T19:20:56Z"  
}
```

11.10.5. Delete property definition

Method	URI	Description
DELETE	/v2/metadefs/namespaces/{namespace}/objects/{object_name}	Deletes an object definition within a namespace.

You cannot delete objects 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: 403

11.10.5.1. Request

This table shows the URI parameters for the delete property definition request:

Name	Type	Description
{namespace}	Uuid	Unique namespace.
{object_name}	String	The name of the object.

This operation does not accept a request body.

11.11. Metadata definition tags (since API v2.0)

Creates, lists, gets details for, updates, and deletes metadata definition tags.

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace}/tags	Creates one or more tag definitions in a namespace.
GET	/v2/metadefs/namespaces/{namespace}/tags{?limit,marker,sort_key,sort_dir}	Lists the tag definitions within a namespace.
DELETE	/v2/metadefs/namespaces/{namespace}/tags	Deletes all tag definitions within a namespace.
POST	/v2/metadefs/namespaces/{namespace}/tags/{name}	Adds a tag to the list of namespace tag definitions.
GET	/v2/metadefs/namespaces/{namespace}/tags/{name}	Gets a definition for a tag.
PUT	/v2/metadefs/namespaces/{namespace}/tags/{name}	Renames a tag definition.
DELETE	/v2/metadefs/namespaces/{namespace}/tags/{name}	Deletes a tag definition within a namespace.

11.11.1. Create tags

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace}/tags	Creates one or more tag definitions in a namespace.

Normal response codes: 201

11.11.1.1. Request

Example 11.42. Create tags: JSON request

```
{
  "tags": [
    {
      "name": "sample-tag1"
    },
    {
      "name": "sample-tag2"
    },
    {
      "name": "sample-tag3"
    }
  ]
}
```

11.11.1.2. Response

Example 11.43. Create tags: JSON response

```
{
  "tags": [
    {
      "name": "sample-tag1"
    },
    {
      "name": "sample-tag2"
    },
    {
      "name": "sample-tag3"
    }
  ]
}
```

11.11.2. List tags

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace}/tags{?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

11.11.2.1. Request

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

Name	Type	Description
<code>limit</code>	Int (<i>Optional</i>)	Requests a page size of returned items from the query. 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.
<code>marker</code>	String (<i>Optional</i>)	Specifies 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.
<code>sort_key</code>	String (<i>Optional</i>)	Sort key. A valid value is an attribute, such as <code>name</code> . for sorting. Default is <code>created_at</code> .
<code>sort_dir</code>	String (<i>Optional</i>)	Sort direction. A valid value is <code>asc</code> (ascending) or <code>desc</code> (descending). Default is <code>desc</code> .

This operation does not accept a request body.

11.11.2.2. Response

Example 11.44. List tags: JSON response

```
{
  "tags": [
    {
      "name": "sample-tag1"
    },
    {
      "name": "sample-tag2"
    },
    {
      "name": "sample-tag3"
    }
  ]
}
```

11.11.3. Delete all tag definitions

Method	URI	Description
DELETE	/v2/metadefs/namespaces/{namespace}/tags	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: 403

11.11.3.1. Request

This operation does not accept a request body.

11.11.4. Add tag definition

Method	URI	Description
POST	/v2/metadefs/namespaces/{namespace}/tags/{name}	Adds a tag to the list of namespace tag definitions.

Normal response codes: 200

11.11.4.1. Request

This operation does not accept a request body.

11.11.4.2. Response

Example 11.45. Add tag definition: JSON response

```
{  
    "created_at": "2015-05-09T01:12:31Z",  
    "name": "added-sample-tag",  
    "updated_at": "2015-05-09T01:12:31Z"  
}
```

11.11.5. Get tag definition

Method	URI	Description
GET	/v2/metadefs/namespaces/{namespace}/tags/{name}	Gets a definition for a tag.

The response body shows a single tag entity.

Normal response codes: 200

11.11.5.1. Request

This operation does not accept a request body.

11.11.5.2. Response

Example 11.46. Get tag definition: JSON response

```
{  
    "created_at": "2015-05-06T23:16:12Z",  
    "name": "sample-tag2",  
    "updated_at": "2015-05-06T23:16:12Z"  
}
```

11.11.6. Update tag definition

Method	URI	Description
PUT	/v2/metadefs/namespaces/{namespace}/tags/{name}	Renames a tag definition.

Normal response codes: 200

11.11.6.1. Request

Example 11.47. Update tag definition: JSON request

```
{  
    "name": "new-tag-name"  
}
```

11.11.6.2. Response

Example 11.48. Update tag definition: JSON response

```
{  
    "name": "new-tag-name"  
}
```

11.11.7. Delete tag definition

Method	URI	Description
DELETE	/v2/metadefs/namespaces/{namespace}/tags/{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: 403

11.11.7.1. Request

This operation does not accept a request body.

11.12. Metadata definition schemas

Gets a JSON-schema document that represents a metadata definition entity.

Method	URI	Description
GET	/v2/schemas/metadefs/namespace	Gets a JSON schema document that represents a metadata definition namespace entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/namespaces	Gets a JSON schema document that represents a metadata definition namespaces entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/object	Gets a JSON schema document that represents a metadata definition object entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/objects	Gets a JSON schema document that represents a metadata definition objects entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/property	Gets a JSON schema document that represents a metadata definition property entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/properties	Gets a JSON schema document that represents a metadata definition properties entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/tag	Gets a JSON schema document that represents a metadata definition tag entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/tags	Gets a JSON schema document that represents a metadata definition tags entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/resource_type	Gets a JSON schema document that represents a metadata definition namespace resource type association entity. (Since API v2.1.)
GET	/v2/schemas/metadefs/resource_types	Gets a JSON schema document that represents a metadata definition namespace resource type associations entity. (Since API v2.1.)

11.12.1. Get metadata definition namespace schema

Method	URI	Description
GET	/v2/schemas/metadefs/namespace	Gets a JSON schema document that represents a metadata definition namespace entity. (Since API v2.1.)

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

11.12.1.1. Request

This operation does not accept a request body.

11.12.1.2. Response

Example 11.49. Get 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",  
                                    "string"  
                                ]  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        "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"
]
}
```

11.12.2. Get metadata definition namespaces schema

Method	URI	Description
GET	/v2/schemas/metadefs/namespaces	Gets a JSON schema document that represents a metadata definition namespaces entity. (Since API v2.1.)

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

11.12.2.1. Request

This operation does not accept a request body.

11.12.2.2. Response

Example 11.50. Get 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"
}
}
}
```

11.12.3. Get metadata definition object schema

Method	URI	Description
GET	/v2/schemas/metadefs/object	Gets a JSON schema document that represents a metadata definition object entity. (Since API v2.1.)

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

11.12.3.1. Request

This operation does not accept a request body.

11.12.3.2. Response

Example 11.51. Get 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",
                                    "object"
                                ]
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```
        "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"
]
}
```

11.12.4. Get metadata definition objects schema

Method	URI	Description
GET	/v2/schemas/metadefs/objects	Gets a JSON schema document that represents a metadata definition objects entity. (Since API v2.1.)

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

11.12.4.1. Request

This operation does not accept a request body.

11.12.4.2. Response

Example 11.52. Get 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"
}
}
}
```

11.12.5. Get metadata definition property schema

Method	URI	Description
GET	/v2/schemas/metadefs/property	Gets a JSON schema document that represents a metadata definition property entity. (Since API v2.1.)

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

11.12.5.1. Request

This operation does not accept a request body.

11.12.5.2. Response

Example 11.53. Get 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"
        },
        "title": {
            "type": "string"
        }
    }
}
```

```
"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"
  ]
}}
```

```
        "object",
        "string",
        null
    ],
    "type": "string"
},
"uniqueItems": {
    "default": false,
    "type": "boolean"
}
},
"required": [
    "type",
    "title",
    "name"
]
}
```

11.12.6. Get metadata definition properties schema

Method	URI	Description
GET	/v2/schemas/metadefs/properties	Gets a JSON schema document that represents a metadata definition properties entity. (Since API v2.1.)

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

11.12.6.1. Request

This operation does not accept a request body.

11.12.6.2. Response

Example 11.54. Get 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"
}
}
}
```

11.12.7. Get metadata definition tag schema

Method	URI	Description
GET	/v2/schemas/metadefs/tag	Gets a JSON schema document that represents a metadata definition tag entity. (Since API v2.1.)

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

11.12.7.1. Request

This operation does not accept a request body.

11.12.7.2. Response

Example 11.55. Get 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"
        }
    }
}
```

11.12.8. Get metadata definition tags schema

Method	URI	Description
GET	/v2/schemas/metadefs/tags	Gets a JSON schema document that represents a metadata definition tags entity. (Since API v2.1.)

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

11.12.8.1. Request

This operation does not accept a request body.

11.12.8.2. Response

Example 11.56. Get 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"
                    }
                }
            }
        }
    }
}
```

```
        "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"
}
}
}
```

11.12.9. Get metadata definition namespace resource type association schema

Method	URI	Description
GET	/v2/schemas/metadefs/resource_type	Gets a JSON schema document that represents a metadata definition namespace resource type association entity. (Since API v2.1.)

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

11.12.9.1. Request

This operation does not accept a request body.

11.12.9.2. Response

Example 11.57. Get 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"
]
}
```

11.12.10. Get metadata definition namespace resource type associations schema

Method	URI	Description
GET	/v2/schemas/metadefs/resource_types	Gets a JSON schema document that represents a metadata definition namespace resource type associations entity. (Since API v2.1.)

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

11.12.10.1. Request

This operation does not accept a request body.

11.12.10.2. Response

Example 11.58. Get 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": {
            "type": "string"
          }
        }
      }
    }
  }
}
```

```

        "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"
}
}
}

```

11.13. Tasks (since API v2.2)

Creates, lists, and gets details for tasks.

Method	URI	Description
POST	/v2/tasks	Creates a task.

Method	URI	Description
GET	/v2/tasks{?type,status,sort_key, sort_dir}	Lists tasks.
GET	/v2/tasks/{task_id}	Shows details for a task.

11.13.1. Create tasks

Method	URI	Description
POST	/v2/tasks	Creates a task.

Normal response codes: 201

11.13.1.1. Request

Example 11.59. Create tasks: JSON request

```
{  
    "type": "import",  
    "input": {  
        "import_from": "http://example.com",  
        "import_from_format": "qcow2",  
        "image_properties": {  
            "disk_format": "vhd",  
            "container_format": "ovf"  
        }  
    }  
}
```

11.13.2. List tasks

Method	URI	Description
GET	/v2/tasks{?type,status,sort_key,sort_dir}	Lists tasks.

Normal response codes: 200

11.13.2.1. Request

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

Name	Type	Description
type	String <i>(Optional)</i>	Filters by a task type. A valid value is import.
status	String <i>(Optional)</i>	Filters by a task status. A valid value is pending, processing, success, or failure.
sort_key	String <i>(Optional)</i>	Sort key. A valid value is an attribute, such as name, for sorting. Default is created_at.
sort_dir	String <i>(Optional)</i>	Sort direction. A valid value is asc (ascending) or desc (descending). Default is desc.

This operation does not accept a request body.

11.13.2.2. Response

Example 11.60. List tasks: JSON response

```
{
  "tasks": [
    {
      "id": "cbc36478b0bd8e67e89469c7749d4127",
      "type": "import",
      "status": "pending"
    },
    {
      "id": "bbc36578b0bd8e67e89469c7749d4126",
      "type": "import",
      "status": "processing"
    }
  ]
}
```

11.13.3. Show tasks

Method	URI	Description
GET	/v2/tasks/{task_id}	Shows details for a task.

Normal response codes: 200

11.13.3.1. Request

This operation does not accept a request body.

11.13.3.2. Response

Example 11.61. Show tasks: JSON response

```
{  
    "id": "e7e59ff6-fa2e-4075-87d3-1a1398a07dc3",  
    "type": "import",  
    "status": "pending"  
}
```

12. 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.
Images		
GET	/v1/images{?name,container_format,disk_format,status,size_min,size_max,changes-since}	Lists 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 details for available images.
GET	/v1/images/{image_id}	Shows the image details as headers and the image binary 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 a member 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 are shared with an owner. The owner ID is the tenant ID.

12.1. API versions

Method	URI	Description
GET	/	Lists information about all Image service API versions.

12.1.1. List API versions

Method	URI	Description
GET	/	Lists information about all Image service API versions.

Normal response codes: 200, 300

12.1.1.1. Request

This operation does not accept a request body.

12.1.1.2. Response

Example 12.1. List API versions: JSON response

```
{
    "versions": [
        {
            "status": "CURRENT",
            "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"
            }
        ]
    }
}
```

12.2. Images

Method	URI	Description
GET	/v1/images{?name,container_format,disk_format,status,size_min,size_max,changes-since}	Lists 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 details for available images.
GET	/v1/images/{image_id}	Shows the image details as headers and the image binary 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.

12.2.1. List images

Method	URI	Description
GET	/v1/images{?name,container_format,disk_format,status,size_min,size_max,changes-since}	Lists public VM images.

Normal response codes: 200

12.2.1.1. Request

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

Name	Type	Description
name	String <i>(Optional)</i>	Filters the image list by an image name, in string format.
container_format	String <i>(Optional)</i>	Filters the image list by a container format, such as ovf, bare, aki, ari, or ami.
disk_format	String <i>(Optional)</i>	Filters the image list by a disk format. A valid value is aki, ari, ami, raw, iso, vhd, vdi, qcow2, or vmdk.
status	String <i>(Optional)</i>	Filters the image list by a status. A valid value is queued, saving, active, killed, deleted, or pending_delete.
size_min	String <i>(Optional)</i>	Filters the image list by a minimum image size, in bytes.
size_max	String <i>(Optional)</i>	Filters the image list by a maximum image size, in bytes.
changes-since	DateTime <i>(Optional)</i>	Filters the image list to those images that have changed since a time stamp value.

This operation does not accept a request body.

12.2.1.2. Response

Example 12.2. 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"
    },
    {
      "...": ...
    }
  ]
}
```

}

12.2.2. Create image

Method	URI	Description
POST	/v1/images	Registers a virtual machine (VM) image.

Normal response codes: 202

12.2.2.1. Request

This operation does not accept a request body.

12.2.2.2. Response

This operation does not return a response body.

12.2.3. List image details

Method	URI	Description
GET	/v1/images/detail{?name, container_format,disk_format,status,size_min,size_max,changes-since}	Lists details for available images.

Normal response codes: 200

12.2.3.1. Request

This table shows the query parameters for the list image details request:

Name	Type	Description
name	String <i>(Optional)</i>	Filters the image list by an image name, in string format.
container_format	String <i>(Optional)</i>	Filters the image list by a container format, such as ovf, bare, aki, ari, or ami.
disk_format	String <i>(Optional)</i>	Filters the image list by a disk format. A valid value is aki, ari, ami, raw, iso, vhd, vdi, qcow2, or vmdk.
status	String <i>(Optional)</i>	Filters the image list by a status. A valid value is queued, saving, active, killed, deleted, or pending_delete.
size_min	String <i>(Optional)</i>	Filters the image list by a minimum image size, in bytes.
size_max	String <i>(Optional)</i>	Filters the image list by a maximum image size, in bytes.
changes-since	DateTime <i>(Optional)</i>	Filters the image list to those images that have changed since a time stamp value.

This operation does not accept a request body.

12.2.3.2. Response

Example 12.3. List image 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"
        }
    },
    {
        "...": ...
    }
]
```

12.2.4. 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)

12.2.4.1. 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.

12.2.4.2. Response

Example 12.4. 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"
      }
    },
    {
      "...": ...
    }
  ]
}
```

12.2.5. 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

12.2.5.1. 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.

12.2.5.2. Response

This operation does not return a response body.

12.2.6. Delete image

Method	URI	Description
DELETE	/v1/images/{image_id}	Deletes an image.

Normal response codes: 204

12.2.6.1. 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.

12.3. 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 a member to an image.
DELETE	/v1/images/{image_id}/members/{owner_id}	Removes a member from an image.

12.3.1. Replace member

Method	URI	Description
PUT	/v1/images/{image_id}/members	Replaces a membership list for an image.

Normal response codes: 204

12.3.1.1. 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.

12.3.2. Add member

Method	URI	Description
PUT	/v1/images/{image_id}/members/{owner_id}	Adds a member 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

12.3.2.1. Request

This table shows the URI parameters for the add 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.

Example 12.5. Add member: JSON request

```
{
  "members": [
    {
      "member_id": "tenant1",
      "can_share": false
    },
    {
      "member_id": "tenant2",
      "can_share": false
    }
  ]
}
```

12.3.3. Remove member

Method	URI	Description
DELETE	/v1/images/{image_id}/members/{owner_id}	Removes a member from an image.

Normal response codes: 204

12.3.3.1. 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.

12.4. Shared images

Method	URI	Description
GET	/v1/shared-images/{owner_id}	Lists the VM images that are shared with an owner. The owner ID is the tenant ID.

12.4.1. List shared images

Method	URI	Description
GET	/v1/shared-images/{owner_id}	Lists the VM images that are shared with an owner. The owner ID is the tenant ID.

Normal response codes: 200

12.4.1.1. 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.

12.4.1.2. Response

Example 12.6. List shared images: JSON response

```
{
  "shared_images": [
    {
      "image_id": "71c675ab-d94f-49cd-a114-e12490b328d9",
      "can_share": false
    },
    {
      "...": ...
    }
  ]
}
```

13. Networking API v2.0 (CURRENT)

Use virtual networking services among devices that are managed by the OpenStack Compute service. 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 a specific IP address from the block or let OpenStack Networking choose the first available IP address.

Effective in the OpenStack Liberty release, XML support in requests and responses was removed for the Networking API v2.0.

Method	URI	Description
API versions		
GET	/	Lists information about all Networking API versions.
GET	/v2.0	Shows details for Networking API v2.0.
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.

13.1. API versions

Lists information for all Networking API versions and shows details about API v2.0.

Method	URI	Description
GET	/	Lists information about all Networking API versions.
GET	/v2.0	Shows details for Networking API v2.0.

13.1.1. List API versions

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

Normal response codes: 200, 300

13.1.1.1. Request

This operation does not accept a request body.

13.1.1.2. Response

Example 13.1. List API versions: JSON response

```
{
    "versions": [
        {
            "status": "CURRENT",
            "id": "v2.0",
            "links": [
                {
                    "href": "http://23.253.228.211:9696/v2.0",
                    "rel": "self"
                }
            ]
        }
    ]
}
```

Example 13.2. List API versions: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<versions xmlns="http://openstack.org/quantum/api/v2.0"
           xmlns:quantum="http://openstack.org/quantum/api/v2.0"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <version>
        <status>CURRENT</status>
        <id>v2.0</id>
        <links>
            <link>
                <href>http://23.253.228.211:9696/v2.0</href>
                <rel>self</rel>
            </link>
        </links>
    </version>
</versions>
```

This operation does not return a response body.

13.1.2. Show API v2.0 details

Method	URI	Description
GET	/v2.0	Shows details for Networking API v2.0.

Normal response codes: 200203

13.1.2.1. Request

This operation does not accept a request body.

13.1.2.2. Response

Example 13.3. Show API v2.0 details: JSON response

```
{
  "resources": [
    {
      "links": [
        {
          "href": "http://23.253.228.211:9696/v2.0/subnets",
          "rel": "self"
        }
      ],
      "name": "subnet",
      "collection": "subnets"
    },
    {
      "links": [
        {
          "href": "http://23.253.228.211:9696/v2.0/networks",
          "rel": "self"
        }
      ],
      "name": "network",
      "collection": "networks"
    },
    {
      "links": [
        {
          "href": "http://23.253.228.211:9696/v2.0/ports",
          "rel": "self"
        }
      ],
      "name": "port",
      "collection": "ports"
    }
  ]
}
```

This table shows the body parameters for the show api v2.0 details response:

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

Example 13.4. Show API v2.0 details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<resources xmlns="http://openstack.org/quantum/api/v2.0"
    xmlns:quantum="http://openstack.org/quantum/api/v2.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <resource>
        <links>
            <link>
                <href>http://23.253.228.211:9696/v2.0/subnets</href>
                <rel>self</rel>
            </link>
        </links>
        <name>subnet</name>
        <collection>subnets</collection>
    </resource>
    <resource>
        <links>
            <link>
                <href>http://23.253.228.211:9696/v2.0/networks</href>
                <rel>self</rel>
            </link>
        </links>
        <name>network</name>
        <collection>networks</collection>
    </resource>
    <resource>
        <links>
            <link>
                <href>http://23.253.228.211:9696/v2.0/ports</href>
                <rel>self</rel>
            </link>
        </links>
        <name>port</name>
        <collection>ports</collection>
    </resource>
</resources>
```

Example 13.5. Show API v2.0 details: JSON response

```
{
    "resources": [
        {
            "links": [
                {
                    "href": "http://23.253.228.211:9696/v2.0/subnets",
                    "rel": "self"
                }
            ],
            "name": "subnet",
            "collection": "subnets"
        },
        {
            "links": [
                {
                    "href": "http://23.253.228.211:9696/v2.0/networks",
                    "rel": "self"
                }
            ],
            "name": "network",
            "collection": "networks"
        }
    ]
}
```

```

        "collection": "networks"
    },
    {
        "links": [
            {
                "href": "http://23.253.228.211:9696/v2.0/ports",
                "rel": "self"
            }
        ],
        "name": "port",
        "collection": "ports"
    }
]
}

```

This table shows the body parameters for the show api v2.0 details response:

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

Example 13.6. Show API v2.0 details: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<resources xmlns="http://openstack.org/quantum/api/v2.0"
           xmlns:quantum="http://openstack.org/quantum/api/v2.0"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <resource>
        <links>
            <link>
                <href>http://23.253.228.211:9696/v2.0/subnets</href>
                <rel>self</rel>
            </link>
        </links>
        <name>subnet</name>
        <collection>subnets</collection>
    </resource>
    <resource>
        <links>
            <link>
                <href>http://23.253.228.211:9696/v2.0/networks</href>
                <rel>self</rel>
            </link>
        </links>
        <name>network</name>
        <collection>networks</collection>
    </resource>
    <resource>
        <links>
            <link>
                <href>http://23.253.228.211:9696/v2.0/ports</href>
                <rel>self</rel>
            </link>
        </links>
        <name>port</name>
        <collection>ports</collection>
    </resource>
</resources>

```

This operation does not return a response body.

13.2. Networks

Lists, shows information 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.

13.2.1. List networks

Method	URI	Description
GET	/v2.0/networks	Lists networks to which the tenant has access.

You can control which response parameters are returned by using the fields query parameter. For information, see [Filtering and column selection](#).

Normal response codes: 200

Error response codes: unauthorized (401)

13.2.1.1. Request

This operation does not accept a request body.

13.2.1.2. Response

Example 13.7. 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,  
            "shared": true,  
            "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",  
            "provider:segmentation_id": null  
        },  
        {  
            "status": "ACTIVE",  
            "subnets": [  
                "08eae331-0402-425a-923c-34f7cf39c1b"  
            ],  
            "name": "private",  
            "provider:physical_network": null,  
            "admin_state_up": true,  
            "tenant_id": "26a7980765d0414dbc1fc1f88cdb7e6e",  
            "provider:network_type": "local",  
            "router:external": true,  
            "shared": true,  
            "id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",  
            "provider:segmentation_id": null  
        }  
    ]  
}
```

Example 13.8. List networks: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<nets xmlns="http://openstack.org/quantum/api/v2.0"
      xmlns:provider="http://docs.openstack.org/ext/provider/api/v1.0"
      xmlns:quantum="http://openstack.org/quantum/api/v2.0"
      xmlns:router="http://docs.openstack.org/ext/neutron/router/api/v1.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <net>
    <status>ACTIVE</status>
    <subnets>
      <subnet>54d6f61d-db07-451c-9ab3-b9609b6b6f0b</subnet>
    </subnets>
    <name>private-network</name>
    <provider:physical_network xsi:nil="true"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
    <provider:network_type>local</provider:network_type>
    <rout:external quantum:type="bool">True</rout:external>
    <shared quantum:type="bool">True</shared>
    <id>d32019d3-bc6e-4319-9c1d-6722fc136a22</id>
    <provider:segmentation_id xsi:nil="true"/>
  </net>
  <net>
    <status>ACTIVE</status>
    <subnets>
      <subnet>08eae331-0402-425a-923c-34f7cf39c1b</subnet>
    </subnets>
    <name>private</name>
    <provider:physical_network xsi:nil="true"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <tenant_id>26a7980765d0414dbc1fc1f88cdb7e6e</tenant_id>
    <provider:network_type>local</provider:network_type>
    <rout:external quantum:type="bool">True</rout:external>
    <shared quantum:type="bool">True</shared>
    <id>db193ab3-96e3-4cb3-8fc5-05f4296d0324</id>
    <provider:segmentation_id xsi:nil="true"/>
  </net>
</nets>
```

13.2.2. Create network

Method	URI	Description
POST	/v2.0/networks	Creates a network.

A request body is optional. An administrative user can specify another tenant ID, which is the tenant who owns the network, in the request body.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

13.2.2.1. Request

Example 13.9. Create network: JSON request

```
{
    "network": {
        "name": "sample_network",
        "admin_state_up": true
    }
}
```

Example 13.10. Create network: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<network>
    <name>sample_network2</name>
</network>
```

13.2.2.2. Response

Example 13.11. Create network: JSON response

```
{
    "network": {
        "status": "ACTIVE",
        "subnets": [],
        "name": "net1",
        "admin_state_up": true,
        "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
        "router:external": false,
        "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"
    }
}
```

Example 13.12. Create network: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<network xmlns="http://openstack.org/quantum/api/v2.0"
          xmlns:provider="http://docs.openstack.org/ext/provider/api/v1.0"
          xmlns:quantum="http://openstack.org/quantum/api/v2.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <status>ACTIVE</status>
    <subnets quantum:type="list"/>
    <name>sample_network2</name>
    <provider:physical_network xsi:nil="true"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
    <provider:network_type>local</provider:network_type>
    <shared quantum:type="bool">False</shared>
    <id>c220b026-ece1-4ead-873f-83537f4c9f92</id>
    <provider:segmentation_id xsi:nil="true"/>
</network>
```

13.2.3. 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)

13.2.3.1. Request

Example 13.13. Bulk create networks: JSON request

```
{
  "networks": [
    {
      "name": "sample_network3",
      "admin_state_up": true
    },
    {
      "name": "sample_network4",
      "admin_state_up": true
    }
  ]
}
```

Example 13.14. Bulk create networks: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<networks>
  <network>
    <name>sample_network_5</name>
  </network>
  <network>
    <name>sample_network_6</name>
  </network>
</networks>
```

13.2.3.2. Response

Example 13.15. Bulk create networks: JSON response

```
{
  "networks": [
    {
      "status": "ACTIVE",
      "subnets": [],
      "name": "sample_network3",
      "provider:physical_network": null,
      "admin_state_up": true,
```

```
        "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
        "provider:network_type": "local",
        "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",
        "provider:network_type": "local",
        "shared": false,
        "id": "af374017-c9ae-4a1d-b799-ab73111476e2",
        "provider:segmentation_id": null
    }
]
}
```

Example 13.16. Bulk create networks: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<networks xmlns="http://openstack.org/quantum/api/v2.0"
           xmlns:provider="http://docs.openstack.org/ext/provider/api/v1.0"
           xmlns:quantum="http://openstack.org/quantum/api/v2.0"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <network>
        <status>ACTIVE</status>
        <subnets quantum:type="list"/>
        <name>sample_network_5</name>
        <provider:physical_network xsi:nil="true"/>
        <admin_state_up quantum:type="bool">True</admin_state_up>
        <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
        <provider:network_type>local</provider:network_type>
        <shared quantum:type="bool">False</shared>
        <id>1f370095-98f6-4079-be64-6d3d4a6adcc6</id>
        <provider:segmentation_id xsi:nil="true"/>
    </network>
    <network>
        <status>ACTIVE</status>
        <subnets quantum:type="list"/>
        <name>sample_network_6</name>
        <provider:physical_network xsi:nil="true"/>
        <admin_state_up quantum:type="bool">True</admin_state_up>
        <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
        <provider:network_type>local</provider:network_type>
        <shared quantum:type="bool">False</shared>
        <id>ee2d3158-3e80-4fb3-ba87-c99f515d85e7</id>
        <provider:segmentation_id xsi:nil="true"/>
    </network>
</networks>
```

13.2.4. 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](#).

In addition to the following response parameters, 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)

13.2.4.1. Request

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

Name	Type	Description
{network_id}	UUID	The UUID for the network of interest to you.

This operation does not accept a request body.

13.2.4.2. Response

Example 13.17. 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",
    "mtu": 0,
    "shared": true,
    "port_security_enabled": true,
    "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22"
  }
}
```

Example 13.18. Show network details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<network xmlns="http://openstack.org/quantum/api/v2.0"
          xmlns:provider="http://docs.openstack.org/ext/provider/api/v1.0"
          xmlns:quantum="http://openstack.org/quantum/api/v2.0"
          xmlns:router="http://docs.openstack.org/ext/neutron/router/api/v1.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <status>ACTIVE</status>
```

```
<subnets>
    <subnet>54d6f61d-db07-451c-9ab3-b9609b6b6f0b</subnet>
</subnets>
<name>private-network</name>
<admin_state_up quantum:type="bool">True</admin_state_up>
<tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
<shared quantum:type="bool">True</shared>
<id>d32019d3-bc6e-4319-9c1d-6722fc136a22</id>
</network>
```

13.2.5. 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)

13.2.5.1. Request

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

Name	Type	Description
{network_id}	UUID	The UUID for the network of interest to you.

Example 13.19. Update network: JSON request

```
{
  "network": {
    "name": "sample_network_5_updated"
  }
}
```

Example 13.20. Update network: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<network xmlns="http://openstack.org/quantum/api/v2.0"
          xmlns:provider="http://docs.openstack.org/ext/provider/api/v1.0"
          xmlns:quantum="http://openstack.org/quantum/api/v2.0"
          xmlns:router="http://docs.openstack.org/ext/quantum/router/api/v1.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <name>sample-network-4-updated</name>
</network>
```

13.2.5.2. Response

Example 13.21. 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",
    "updated_at": "2015-11-17T14:44:29.000000Z",
    "created_at": "2015-11-17T14:44:29.000000Z",
    "extra_dhcp_opts": []
  }
}
```

```
        "provider:segmentation_id": null
    }
}
```

Example 13.22. Update network: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<network xmlns="http://openstack.org/quantum/api/v2.0"
          xmlns:provider="http://docs.openstack.org/ext/provider/api/v1.0"
          xmlns:quantum="http://openstack.org/quantum/api/v2.0"
          xmlns:router="http://docs.openstack.org/ext/neutron/router/api/v1.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <status>ACTIVE</status>
    <subnets quantum:type="list"/>
    <name>sample-network-4-updated</name>
    <provider:physical_network xsi:nil="true"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
    <provider:network_type>local</provider:network_type>
    <router:external quantum:type="bool">False</router:external>
    <shared quantum:type="bool">False</shared>
    <id>af374017-c9ae-4a1d-b799-ab73111476e2</id>
    <provider:segmentation_id xsi:nil="true"/>
</network>
```

13.2.6. 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)

13.2.6.1. Request

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

Name	Type	Description
{network_id}	UUID	The UUID for the network of interest to you.

This operation does not accept a request body.

13.3. Subnets

Lists, shows information 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.

13.3.1. 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)

13.3.1.1. Request

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

Name	Type	Description
display_name	String <i>(Optional)</i>	The name of the network.
network_id	Uuid <i>(Optional)</i>	The UUID of the attached network.
gateway_ip	String <i>(Optional)</i>	The gateway IP address.
ip_version	Int <i>(Optional)</i>	The IP version, which is 4 or 6.
cidr	Bool <i>(Optional)</i>	The CIDR.
id	Uuid <i>(Optional)</i>	The UUID of the subnet.
enable_dhcp	Boolean <i>(Optional)</i>	If true, DHCP is enabled. If false, DHCP is disabled.
ipv6_ra_mode	String <i>(Optional)</i>	Choose from constants.IPV6_SLAAC, constants.DHCPV6_STATEFUL, constants.DHCPV6_STATELESS, name='ipv6_address_modes, or null.
ipv6_address_mode	String <i>(Optional)</i>	Choose from constants.IPV6_SLAAC, constants.DHCPV6_STATEFUL, constants.DHCPV6_STATELESS, name='ipv6_address_modes, or null.

13.3.1.2. Response

Example 13.23. 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-34f7cf39c1b"
    },
    {
        "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"
    }
]
}

```

Example 13.24. List subnets: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<subnets>
    <subnet>
        <name>private-subnet</name>
        <enable_dhcp>True</enable_dhcp>
        <network_id>db193ab3-96e3-4cb3-8fc5-05f4296d0324</network_id>
        <tenant_id>26a7980765d0414dbc1fc1f88cdb7e6e</tenant_id>
        <dns_nameservers/>
        <allocation_pools>
            <allocation_pool>
                <start>10.0.0.2</start>
                <end>10.0.0.254</end>
            </allocation_pool>
        </allocation_pools>
        <host_routes/>
        <ip_version>4</ip_version>
        <gateway_ip>10.0.0.1</gateway_ip>
        <cidr>10.0.0.0/24</cidr>
    </subnet>

```

```
<id>08eae331-0402-425a-923c-34f7cf39c1b</id>
</subnet>
<subnet>
    <name>my_subnet</name>
    <enable_dhcp>True</enable_dhcp>
    <network_id>d32019d3-bc6e-4319-9c1d-6722fc136a22</network_id>
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
    <dns_nameservers/>
    <allocation_pools>
        <allocation_pool>
            <start>192.0.0.2</start>
            <end>192.255.255.254</end>
        </allocation_pool>
    </allocation_pools>
    <host_routes/>
    <ip_version>4</ip_version>
    <gateway_ip>192.0.0.1</gateway_ip>
    <cidr>192.0.0.0/8</cidr>
    <id>54d6f61d-db07-451c-9ab3-b9609b6b6f0b</id>
</subnet>
</subnets>
```

This operation does not return a response body.

13.3.2. 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)

13.3.2.1. Request

Example 13.25. Create subnet: JSON request

```
{
    "subnet": {
        "network_id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
        "ip_version": 4,
        "cidr": "10.0.0.1"
    }
}
```

Example 13.26. Create subnet: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<subnet>
    <network_id>d32019d3-bc6e-4319-9c1d-6722fc136a22</network_id>
    <ip_version>4</ip_version>
    <cidr>10.0.0.1</cidr>
</subnet>
```

This operation does not accept a request body.

13.3.2.2. Response

Example 13.27. 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"  
    }  
}
```

Example 13.28. Create subnet: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<subnet>  
    <name>test_subnet_1</name>  
    <enable_dhcp>True</enable_dhcp>  
    <network_id>d32019d3-bc6e-4319-9c1d-6722fc136a22</network_id>  
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>  
    <dns_nameservers/>  
    <allocation_pools>  
        <allocation_pool>  
            <start>192.0.0.2</start>  
            <end>192.255.255.254</end>  
        </allocation_pool>  
    </allocation_pools>  
    <host_routes/>  
    <ip_version>4</ip_version>  
    <gateway_ip>192.0.0.1</gateway_ip>  
    <cidr>192.0.0.0/8</cidr>  
    <id>54d6f61d-db07-451c-9ab3-b9609b6b6f0b</id>  
</subnet>
```

This operation does not return a response body.

13.3.3. 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)

13.3.3.1. Request

Example 13.29. 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"
    }
  ]
}
```

Example 13.30. Bulk create subnet: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<subnets>
  <subnet>
    <name>test_subnet_1</name>
    <network_id>a3775a7d-9f8b-4148-be81-c84bbd0837ce</network_id>
    <cidr>10.0.0.0/8</cidr>
    <ip_version>4</ip_version>
  </subnet>
  <subnet>
    <name>test_subnet_2</name>
    <network_id>a3775a7d-9f8b-4148-be81-c84bbd0837ce</network_id>
    <cidr>192.168.0.0/16</cidr>
    <ip_version>4</ip_version>
  </subnet>
</subnets>
```

This operation does not accept a request body.

13.3.3.2. Response

Example 13.31. 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"
    }
  ]
}
```

Example 13.32. Bulk create subnet: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<subnets>
  <subnet>
    <name>test_subnet_1</name>
    <enable_dhcp>True</enable_dhcp>
    <network_id>a3775a7d-9f8b-4148-be81-c84bbd0837ce</network_id>
    <tenant_id>60cd4f6dbc2f491982a284e7b83b5be3</tenant_id>
    <dns_nameservers/>
    <allocation_pools>
      <allocation_pool>
        <start>10.0.0.2</start>
        <end>10.255.255.254</end>
      
```

```
        </allocation_pool>
    </allocation_pools>
    <host_routes/>
    <ip_version>4</ip_version>
    <gateway_ip>10.0.0.1</gateway_ip>
    <cidr>10.0.0.0/8</cidr>
    <id>bd3fd365-fe19-431a-be63-07017a09316c</id>
</subnet>
<subnet>
    <name>test_subnet_2</name>
    <enable_dhcp>True</enable_dhcp>
    <network_id>a3775a7d-9f8b-4148-be81-c84bb0837ce</network_id>
    <tenant_id>60cd4f6dbc2f491982a284e7b83b5be3</tenant_id>
    <dns_nameservers/>
    <allocation_pools>
        <allocation_pool>
            <start>192.168.0.2</start>
            <end>192.168.255.254</end>
        </allocation_pool>
    </allocation_pools>
    <host_routes/>
    <ip_version>4</ip_version>
    <gateway_ip>192.168.0.1</gateway_ip>
    <cidr>192.168.0.0/16</cidr>
    <id>86e7c838-fb75-402b-9dbf-d68166e3f5fe</id>
</subnet>
</subnets>
```

This operation does not return a response body.

13.3.4. 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)

13.3.4.1. Request

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

Name	Type	Description
{subnet_id}	UUID	The UUID for the subnet of interest to you.

This operation does not accept a request body.

13.3.4.2. Response

Example 13.33. Show subnet details: JSON response

```
{
    "subnet": {
        "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"
    }
}
```

Example 13.34. Show subnet details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<subnet>
    <name>test_subnet_1</name>
    <enable_dhcp>True</enable_dhcp>
    <network_id>d32019d3-bc6e-4319-9c1d-6722fc136a22</network_id>
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
    <dns_nameservers/>
    <allocation_pools>
```

```
<allocation_pool>
  <start>192.0.0.2</start>
  <end>192.255.255.254</end>
</allocation_pool>
</allocation_pools>
<host_routes/>
<ip_version>4</ip_version>
<gateway_ip>192.0.0.1</gateway_ip>
<cidr>192.0.0.0/8</cidr>
<id>54d6f61d-db07-451c-9ab3-b9609b6b6f0b</id>
</subnet>
```

This operation does not return a response body.

13.3.5. 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)

13.3.5.1. Request

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

Name	Type	Description
{subnet_id}	UUID	The UUID for the subnet of interest to you.

Example 13.35. Update subnet: JSON request

```
{
    "subnet": {
        "name": "my_subnet"
    }
}
```

Example 13.36. Update subnet: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<subnet>
    <name>my_subnet</name>
</subnet>
```

This operation does not accept a request body.

13.3.5.2. Response

Example 13.37. 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"
            }
        ],
        "cidr": "10.0.0.0/24"
    }
}
```

```
        "host_routes": [],
        "ip_version": 4,
        "gateway_ip": "10.0.0.1",
        "cidr": "10.0.0.0/24",
        "id": "08eae331-0402-425a-923c-34f7cf39c1b"
    }
}
```

Example 13.38. Update subnet: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<subnet>
    <name>my_subnet</name>
    <enable_dhcp>True</enable_dhcp>
    <network_id>d32019d3-bc6e-4319-9c1d-6722fc136a22</network_id>
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
    <dns_nameservers/>
    <allocation_pools>
        <allocation_pool>
            <start>192.0.0.2</start>
            <end>192.255.255.254</end>
        </allocation_pool>
    </allocation_pools>
    <host_routes/>
    <ip_version>4</ip_version>
    <gateway_ip>192.0.0.1</gateway_ip>
    <cidr>192.0.0.0/8</cidr>
    <id>54d6f61d-db07-451c-9ab3-b9609b6b6f0b</id>
</subnet>
```

This operation does not return a response body.

13.3.6. 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)

13.3.6.1. Request

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

Name	Type	Description
{subnet_id}	UUID	The UUID for the subnet of interest to you.

This operation does not accept a request body.

13.4. Ports

Lists, shows information 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.

13.4.1. 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. 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)

13.4.1.1. Request

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

Name	Type	Description
status	String <i>(Optional)</i>	The port status. Value is ACTIVE or DOWN.
display_name	String <i>(Optional)</i>	The port name.
admin_state	Bool <i>(Optional)</i>	The administrative state of the router, which is up (true) or down (false).
network_id	Uuid <i>(Optional)</i>	The UUID of the attached network.
tenant_id	Uuid <i>(Optional)</i>	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 <i>(Optional)</i>	The UUID of the entity that uses this port. For example, a DHCP agent.
mac_address	String <i>(Optional)</i>	The MAC address of the port.
port_id	Uuid <i>(Optional)</i>	The UUID of the port.
security_groups	Uuid <i>(Optional)</i>	The UUIDs of any attached security groups.
device_id	Uuid <i>(Optional)</i>	The UUID of the device that uses this port. For example, a virtual server.

13.4.1.2. Response

Example 13.39. 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"
    }
  ]
}
```

Example 13.40. List ports: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<ports xmlns="http://openstack.org/quantum/api/v2.0"
       xmlns:quantum="http://openstack.org/quantum/api/v2.0"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <port>
    <status>ACTIVE</status>
    <name/>
    <allowed_address_pairs quantum:type="list"/>
```

```
<admin_state_up quantum:type="bool">True</admin_state_up>
<network_id>70c1db1f-b701-45bd-96e0-a313ee3430b3</network_id>
<tenant_id/>
<extra_dhcp_opts quantum:type="list"/>
<device_owner>network:router_gateway</device_owner>
<mac_address>fa:16:3e:58:42:ed</mac_address>
<fixed_ips>
    <fixed_ip>
        <subnet_id>008ba151-0b8c-4a67-98b5-0d2b87666062</subnet_id>
        <ip_address>172.24.4.2</ip_address>
    </fixed_ip>
</fixed_ips>
<id>d80b1a3b-4fc1-49f3-952e-1e2ab7081d8b</id>
<security_groups quantum:type="list"/>
<device_id>9ae135f4-b6e0-4dad-9e91-3c223e385824</device_id>
</port>
<port>
    <status>ACTIVE</status>
    <name/>
    <allowed_address_pairs quantum:type="list"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <network_id>f27aa545-cbdd-4907-b0c6-c9e8b039dcc2</network_id>
    <tenant_id>d397de8a63f341818f198abb0966f6f3</tenant_id>
    <extra_dhcp_opts quantum:type="list"/>
    <device_owner>network:router_interface</device_owner>
    <mac_address>fa:16:3e:bb:3c:e4</mac_address>
    <fixed_ips>
        <fixed_ip>
            <subnet_id>288bf4a1-51ba-43b6-9d0a-520e9005db17</subnet_id>
            <ip_address>10.0.0.1</ip_address>
        </fixed_ip>
    </fixed_ips>
    <id>f71a6703-d6de-4be1-a91a-a570ede1d159</id>
    <security_groups quantum:type="list"/>
    <device_id>9ae135f4-b6e0-4dad-9e91-3c223e385824</device_id>
</port>
</ports>
```

This operation does not return a response body.

13.4.2. Create port

Method	URI	Description
POST	/v2.0/ports	Creates a port on a network.

You must specify the `network_id` attribute in the request body to define the network where the port is to be created.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), macGenerationFailure (503), serviceUnavailable (503)

13.4.2.1. Request

Example 13.41. Create port: JSON request

```
{
  "port": {
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "name": "private-port",
    "admin_state_up": true
  }
}
```

Example 13.42. Create port: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<port>
  <name>test_port_1</name>
  <network_id>a87cc70a-3e15-4acf-8205-9b711a3531b7</network_id>
</port>
```

This operation does not accept a request body.

13.4.2.2. Response

Example 13.43. 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": "a87cc70a-3e15-4acf-8205-9b711a3531b7"
  }
}
```

```
        "id": "65c0ee9f-d634-4522-8954-51021b570b0d",
        "security_groups": [
            "f0ac4394-7e4a-4409-9701-ba8be283dbc3"
        ],
        "device_id": ""
    }
}
```

Example 13.44. Create port: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<port xmlns="http://openstack.org/quantum/api/v2.0"
      xmlns:quantum="http://openstack.org/quantum/api/v2.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <status>DOWN</status>
    <name>test_port_1</name>
    <allowed_address_pairs quantum:type="list"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <network_id>a87cc70a-3e15-4acf-8205-9b711a3531b7</network_id>
    <tenant_id>d6700c0c9ffa4f1cb322cd4a1f3906fa</tenant_id>
    <device_owner/>
    <mac_address>fa:16:3e:09:e3:47</mac_address>
    <fixed_ips>
        <fixed_ip>
            <subnet_id>a0304c3a-4f08-4c43-88af-d796509c97d2</subnet_id>
            <ip_address>10.0.0.4</ip_address>
        </fixed_ip>
    </fixed_ips>
    <id>8021790b-4bfd-46ab-bcc7-0ef2f73bff43</id>
    <security_groups>
        <security_group>f0ac4394-7e4a-4409-9701-ba8be283dbc3</security_group>
    </security_groups>
    <device_id/>
</port>
```

This operation does not return a response body.

13.4.3. 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)

13.4.3.1. Request

Example 13.45. 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"
    }
  ]
}
```

Example 13.46. Bulk create ports: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<ports>
  <port>
    <name>test_port_1-xml</name>
    <network_id>a87cc70a-3e15-4acf-8205-9b711a3531b7</network_id>
  </port>
  <port>
    <name>test_port_2-xml</name>
    <network_id>a87cc70a-3e15-4acf-8205-9b711a3531b7</network_id>
  </port>
</ports>
```

This operation does not accept a request body.

13.4.3.2. Response

Example 13.47. 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": ""
}
]
}

```

Example 13.48. Bulk create ports: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<ports xmlns="http://openstack.org/quantum/api/v2.0"
       xmlns:quantum="http://openstack.org/quantum/api/v2.0"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <port>
        <status>DOWN</status>
        <name>test_port_1-xml</name>
        <allowed_address_pairs quantum:type="list"/>
        <admin_state_up quantum:type="bool">True</admin_state_up>
        <network_id>a87cc70a-3e15-4acf-8205-9b711a3531b7</network_id>
        <tenant_id>d6700c0c9ffa4f1cb322cd4a1f3906fa</tenant_id>
        <device_owner/>
        <mac_address>fa:16:3e:fa:e2:34</mac_address>
        <fixed_ips>

```

```
<fixed_ip>
    <subnet_id>a0304c3a-4f08-4c43-88af-d796509c97d2</subnet_id>
    <ip_address>10.0.0.7</ip_address>
</fixed_ip>
</fixed_ips>
<id>054e8f14-4082-400e-afcc-5d6e5b3bcc0c</id>
<security_groups>
    <security_group>f0ac4394-7e4a-4409-9701-ba8be283dbc3</
security_group>
    </security_groups>
    <device_id/>
</port>
<port>
    <status>DOWN</status>
    <name>test_port_2-xml</name>
    <allowed_address_pairs quantum:type="list"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <network_id>a87cc70a-3e15-4acf-8205-9b711a3531b7</network_id>
    <tenant_id>d6700c0c9ffa4f1cb322cd4a1f3906fa</tenant_id>
    <device_owner/>
    <mac_address>fa:16:3e:e6:cf:d9</mac_address>
    <fixed_ips>
        <fixed_ip>
            <subnet_id>a0304c3a-4f08-4c43-88af-d796509c97d2</subnet_id>
            <ip_address>10.0.0.8</ip_address>
        </fixed_ip>
    </fixed_ips>
    <id>879e96f9-6dd5-4232-bd19-3f39d0ae463b</id>
    <security_groups>
        <security_group>f0ac4394-7e4a-4409-9701-ba8be283dbc3</
security_group>
        </security_groups>
        <device_id/>
    </port>
</ports>
```

This operation does not return a response body.

13.4.4. 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)

13.4.4.1. Request

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

Name	Type	Description
{port_id}	UUID	The UUID for the port of interest to you.

This operation does not accept a request body.

13.4.4.2. Response

Example 13.49. 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",
    "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-e6dc1ccedb2",
    "security_groups": [],
    "device_id": "5e3898d7-11be-483e-9732-b2f5eccd2b2e"
  }
}
```

Example 13.50. Show port details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<port xmlns="http://openstack.org/quantum/api/v2.0"
      xmlns:quantum="http://openstack.org/quantum/api/v2.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <status>ACTIVE</status>
  <name/>
  <allowed_address_pairs quantum:type="list"/>
  <admin_state_up quantum:type="bool">True</admin_state_up>
```

```
<network_id>a87cc70a-3e15-4acf-8205-9b711a3531b7</network_id>
<tenant_id>7e02058126cc4950b75f9970368ba177</tenant_id>
<extra_dhcp_opts quantum:type="list"/>
<device_owner>network:router_interface</device_owner>
<mac_address>fa:16:3e:23:fd:d7</mac_address>
<fixed_ips>
    <fixed_ip>
        <subnet_id>a0304c3a-4f08-4c43-88af-d796509c97d2</subnet_id>
        <ip_address>10.0.0.1</ip_address>
    </fixed_ip>
</fixed_ips>
<id>46d4bfb9-b26e-41f3-bd2e-e6dcc1ccedb2</id>
<security_groups quantum:type="list"/>
<device_id>5e3898d7-11be-483e-9732-b2f5eccd2b2e</device_id>
</port>
```

This operation does not return a response body.

13.4.5. 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)

13.4.5.1. Request

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

Name	Type	Description
{port_id}	UUID	The UUID for the port of interest to you.

Example 13.51. 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"
  }
}
```

Example 13.52. Update port: JSON request

```
<?xml version="1.0" encoding="UTF-8"?>
<port>
  <name>test-for-port-update</name>
  <device_owner>compute:nova</device_owner>
</port>
```

This operation does not accept a request body.

13.4.5.2. Response

Example 13.53. 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"
  }
}
```

Example 13.54. Update port: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<port xmlns="http://openstack.org/quantum/api/v2.0"
      xmlns:quantum="http://openstack.org/quantum/api/v2.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <status>DOWN</status>
  <name>test-for-port-update</name>
  <allowed_address_pairs quantum:type="list"/>
  <admin_state_up quantum:type="bool">True</admin_state_up>
  <network_id>a87cc70a-3e15-4acf-8205-9b711a3531b7</network_id>
  <tenant_id>d6700c0c9ffa4f1cb322cd4a1f3906fa</tenant_id>
  <device_owner>compute:nova</device_owner>
  <mac_address>fa:16:3e:11:11:5e</mac_address>
  <fixed_ips>
    <fixed_ip>
      <subnet_id>898dec4a-74df-4193-985f-c76721bcc746</subnet_id>
      <ip_address>20.20.0.4</ip_address>
    </fixed_ip>
  </fixed_ips>
  <id>43c831e0-19ce-4a76-9a49-57b57e69428b</id>
  <security_groups>
    <security_group>ce0179d6-8a94-4f7c-91c2-f3038e2acbd0</security_group>
  </security_groups>
  <device_id/>

```

```
</port>
```

This operation does not return a response body.

13.4.6. 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)

13.4.6.1. Request

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

Name	Type	Description
{port_id}	UUID	The UUID for the port of interest to you.

This operation does not accept a request body.

13.5. Service providers

Lists service providers.

Method	URI	Description
GET	/v2.0/service-providers	Lists service providers.

13.5.1. List service providers

Method	URI	Description
GET	/v2.0/service-providers	Lists 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)

13.5.1.1. Request

This operation does not accept a request body.

13.5.1.2. Response

Example 13.55. List service providers: JSON response

```
{  
    "service_providers": [  
        {  
            "service_type": "LOADBALANCER",  
            "default": true,  
            "name": "haproxy"  
        }  
    ]  
}
```

14. Networking API v2.0 extensions (CURRENT)

Method	URI	Description
Extensions		
GET	/v2.0/extensions	Lists available Networking API extensions.
GET	/v2.0/extensions/{alias}	Shows details for an extension.
Quotas extension (quotas)		
GET	/v2.0/quotas	Lists quotas for tenants who have non-default quota values.
GET	/v2.0/quotas/{tenant_id}	Shows 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.
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)		
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.
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.

Method	URI	Description
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/{rules-security-groups-id}	Shows detailed information for a security group rule.
DELETE	/v2.0/security-group-rules/{rules-security-groups-id}	Deletes a rule from an OpenStack Networking security group.
Layer-3 networking		
GET	/v2.0/routers	Lists logical routers that are accessible to the tenant who submits the request.
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	Removes 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.
Metering labels and 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 information 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 detailed information for a metering label rule.
DELETE	/v2.0/metering/metering-label-rules/{metering-label-rule-id}	Deletes a L3 metering label rule.
Firewall-as-a-Service (FWaaS) 2.0 (CURRENT)		
GET	/v2.0/fw/firewalls	List firewalls.
POST	/v2.0/fw/firewalls	Creates a firewall.

Method	URI	Description
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}	Removes a firewall.
Load-Balancer-as-a-Service (LBaaS) 1.0 (STABLE)		
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.
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.
Load-Balancer-as-a-Service (LBaaS) 2.0 (EXPERIMENTAL)		
GET	/v2.0/lbaas/loadbalancers	Lists load balancers.
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/listeners	Lists 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.

Method	URI	Description
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 of 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/{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.
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.
Virtual-Private-Network-as-a-Service (VPNaaS) 2.0 (CURRENT)		
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.

Method	URI	Description
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.
Extra routes		
PUT	/v2.0/routers/{router_id}	Configures extra routes on a router.

14.1. Extensions

Lists available Networking API v2.0 extensions and shows details for an extension.

Method	URI	Description
GET	/v2.0/extensions	Lists available Networking API extensions.
GET	/v2.0/extensions/{alias}	Shows details for an extension.

14.1.1. List extensions

Method	URI	Description
GET	/v2.0/extensions	Lists available Networking API extensions.

Normal response codes: 200203

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

14.1.1.1. Request

This operation does not accept a request body.

14.1.1.2. Response

Example 14.1. 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"
}
]
}

```

Example 14.2. List extensions: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<extensions xmlns="http://openstack.org/quantum/api/v2.0"
             xmlns:quantum="http://openstack.org/quantum/api/v2.0"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <extension>
        <updated>2013-01-20T00:00:00-00:00</updated>
        <name>Neutron Service Type Management</name>
        <links quantum:type="list"/>
        <alias>service-type</alias>
        <description>API for retrieving service providers for Neutron  
advanced services</description>
    </extension>
    <extension>
        <updated>2012-10-05T10:00:00-00:00</updated>
        <name>security-group</name>
        <links quantum:type="list"/>
        <alias>security-group</alias>
        <description>The security groups extension.</description>
    </extension>
    <extension>
        <updated>2013-02-07T10:00:00-00:00</updated>
        <name>L3 Agent Scheduler</name>
        <links quantum:type="list"/>
        <alias>l3_agent_scheduler</alias>
    </extension>

```

```
<description>Schedule routers among 13 agents</description>
</extension>
<extension>
    <updated>2013-02-07T10:00:00-00:00</updated>
    <name>Loadbalancer Agent Scheduler</name>
    <links quantum:type="list"/>
    <alias>lbaas_agent_scheduler</alias>
    <description>Schedule pools among lbaas agents</description>
</extension>
<extension>
    <updated>2013-03-28T10:00:00-00:00</updated>
    <name>Neutron L3 Configurable external gateway mode</name>
    <links quantum:type="list"/>
    <alias>ext-gw-mode</alias>
    <description>Extension of the router abstraction for
        specifying whether SNAT should occur on the external
        gateway</description>
</extension>
<extension>
    <updated>2014-02-03T10:00:00-00:00</updated>
    <name>Port Binding</name>
    <links quantum:type="list"/>
    <alias>binding</alias>
    <description>Expose port bindings of a virtual port to
        external application</description>
</extension>
<extension>
    <updated>2012-09-07T10:00:00-00:00</updated>
    <name>Provider Network</name>
    <links quantum:type="list"/>
    <alias>provider</alias>
    <description>Expose mapping of virtual networks to physical
        networks</description>
</extension>
<extension>
    <updated>2013-02-03T10:00:00-00:00</updated>
    <name>agent</name>
    <links quantum:type="list"/>
    <alias>agent</alias>
    <description>The agent management extension.</description>
</extension>
<extension>
    <updated>2012-07-29T10:00:00-00:00</updated>
    <name>Quota management support</name>
    <links quantum:type="list"/>
    <alias>quotas</alias>
    <description>Expose functions for quotas management per
        tenant</description>
</extension>
<extension>
    <updated>2013-02-07T10:00:00-00:00</updated>
    <name>DHCP Agent Scheduler</name>
    <links quantum:type="list"/>
    <alias>dhcp_agent_scheduler</alias>
    <description>Schedule networks among dhcp agents</description>
</extension>
<extension>
    <updated>2013-06-27T10:00:00-00:00</updated>
    <name>Multi Provider Network</name>
    <links quantum:type="list"/>
```

```

<alias>multi-provider</alias>
<description>Expose mapping of virtual networks to multiple
physical networks</description>
</extension>
<extension>
<updated>2013-01-14T10:00:00-00:00</updated>
<name>Neutron external network</name>
<links quantum:type="list"/>
<alias>external-net</alias>
<description>Adds external network attribute to network
resource.</description>
</extension>
<extension>
<updated>2012-07-20T10:00:00-00:00</updated>
<name>Neutron L3 Router</name>
<links quantum:type="list"/>
<alias>router</alias>
<description>Router abstraction for basic L3 forwarding
between L2 Neutron networks and access to external
networks via a NAT gateway.</description>
</extension>
<extension>
<updated>2013-07-23T10:00:00-00:00</updated>
<name>Allowed Address Pairs</name>
<links quantum:type="list"/>
<alias>allowed-address-pairs</alias>
<description>Provides allowed address pairs</description>
</extension>
<extension>
<updated>2013-03-17T12:00:00-00:00</updated>
<name>Neutron Extra DHCP opts</name>
<links quantum:type="list"/>
<alias>extra_dhcp_opt</alias>
<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)</description>
</extension>
<extension>
<updated>2012-10-07T10:00:00-00:00</updated>
<name>LoadBalancing service</name>
<links quantum:type="list"/>
<alias>lbaas</alias>
<description>Extension for LoadBalancing service</description>
</extension>
<extension>
<updated>2013-02-01T10:00:00-00:00</updated>
<name>Neutron Extra Route</name>
<links quantum:type="list"/>
<alias>extraroute</alias>
<description>Extra routes configuration for L3
router</description>
</extension>
</extensions>

```

Example 14.3. 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"  
}  
]  
}
```

Example 14.4. List extensions: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<extensions xmlns="http://openstack.org/quantum/api/v2.0"  
            xmlns:quantum="http://openstack.org/quantum/api/v2.0"  
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
    <extension>  
        <updated>2013-01-20T00:00:00-00:00</updated>  
        <name>Neutron Service Type Management</name>  
        <links quantum:type="list"/>  
        <alias>service-type</alias>  
        <description>API for retrieving service providers for Neutron  
                advanced services</description>  
    </extension>  
    <extension>  
        <updated>2012-10-05T10:00:00-00:00</updated>  
        <name>security-group</name>  
        <links quantum:type="list"/>  
        <alias>security-group</alias>  
        <description>The security groups extension.</description>  
    </extension>  
    <extension>  
        <updated>2013-02-07T10:00:00-00:00</updated>  
        <name>L3 Agent Scheduler</name>  
        <links quantum:type="list"/>  
        <alias>l3_agent_scheduler</alias>  
        <description>Schedule routers among 13 agents</description>  
    </extension>  
    <extension>  
        <updated>2013-02-07T10:00:00-00:00</updated>  
        <name>Loadbalancer Agent Scheduler</name>  
        <links quantum:type="list"/>  
        <alias>lbaas_agent_scheduler</alias>  
        <description>Schedule pools among lbaas agents</description>  
    </extension>  
    <extension>  
        <updated>2013-03-28T10:00:00-00:00</updated>  
        <name>Neutron L3 Configurable external gateway mode</name>  
        <links quantum:type="list"/>  
        <alias>ext-gw-mode</alias>  
        <description>Extension of the router abstraction for  
                specifying whether SNAT should occur on the external  
                gateway</description>  
    </extension>  
    <extension>  
        <updated>2014-02-03T10:00:00-00:00</updated>  
        <name>Port Binding</name>  
        <links quantum:type="list"/>  
        <alias>binding</alias>  
        <description>Expose port bindings of a virtual port to
```

```
        external application</description>
    </extension>
    <extension>
        <updated>2012-09-07T10:00:00-00:00</updated>
        <name>Provider Network</name>
        <links quantum:type="list"/>
        <alias>provider</alias>
        <description>Expose mapping of virtual networks to physical
            networks</description>
    </extension>
    <extension>
        <updated>2013-02-03T10:00:00-00:00</updated>
        <name>agent</name>
        <links quantum:type="list"/>
        <alias>agent</alias>
        <description>The agent management extension.</description>
    </extension>
    <extension>
        <updated>2012-07-29T10:00:00-00:00</updated>
        <name>Quota management support</name>
        <links quantum:type="list"/>
        <alias>quotas</alias>
        <description>Expose functions for quotas management per
            tenant</description>
    </extension>
    <extension>
        <updated>2013-02-07T10:00:00-00:00</updated>
        <name>DHCP Agent Scheduler</name>
        <links quantum:type="list"/>
        <alias>dhcp_agent_scheduler</alias>
        <description>Schedule networks among dhcp agents</description>
    </extension>
    <extension>
        <updated>2013-06-27T10:00:00-00:00</updated>
        <name>Multi Provider Network</name>
        <links quantum:type="list"/>
        <alias>multi-provider</alias>
        <description>Expose mapping of virtual networks to multiple
            physical networks</description>
    </extension>
    <extension>
        <updated>2013-01-14T10:00:00-00:00</updated>
        <name>Neutron external network</name>
        <links quantum:type="list"/>
        <alias>external-net</alias>
        <description>Adds external network attribute to network
            resource.</description>
    </extension>
    <extension>
        <updated>2012-07-20T10:00:00-00:00</updated>
        <name>Neutron L3 Router</name>
        <links quantum:type="list"/>
        <alias>router</alias>
        <description>Router abstraction for basic L3 forwarding
            between L2 Neutron networks and access to external
            networks via a NAT gateway.</description>
    </extension>
    <extension>
        <updated>2013-07-23T10:00:00-00:00</updated>
        <name>Allowed Address Pairs</name>
```

```
<links quantum:type="list"/>
<alias>allowed-address-pairs</alias>
<description>Provides allowed address pairs</description>
</extension>
<extension>
<updated>2013-03-17T12:00:00-00:00</updated>
<name>Neutron Extra DHCP opts</name>
<links quantum:type="list"/>
<alias>extra_dhcp_opt</alias>
<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)</description>
</extension>
<extension>
<updated>2012-10-07T10:00:00-00:00</updated>
<name>LoadBalancing service</name>
<links quantum:type="list"/>
<alias>lbaas</alias>
<description>Extension for LoadBalancing service</description>
</extension>
<extension>
<updated>2013-02-01T10:00:00-00:00</updated>
<name>Neutron Extra Route</name>
<links quantum:type="list"/>
<alias>extraroute</alias>
<description>Extra routes configuration for L3
router</description>
</extension>
</extensions>
```

This operation does not return a response body.

14.1.2. Show extension details

Method	URI	Description
GET	/v2.0/extensions/{alias}	Shows details for an extension.

Normal response codes: 200203

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

14.1.2.1. 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.

14.1.2.2. Response

Example 14.5. Get extension details: JSON response

```
{
  "extension": {
    "updated": "2013-02-03T10:00:00-00:00",
    "name": "agent",
    "links": [],
    "alias": "agent",
    "description": "The agent management extension."
  }
}
```

Example 14.6. Get extension details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extension xmlns="http://openstack.org/quantum/api/v2.0"
           xmlns:quantum="http://openstack.org/quantum/api/v2.0"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <updated>2013-02-03T10:00:00-00:00</updated>
  <name>agent</name>
  <links quantum:type="list"/>
  <alias>agent</alias>
  <description>The agent management extension.</description>
</extension>
```

Example 14.7. Get extension details: JSON response

```
{
  "extension": {
    "updated": "2013-02-03T10:00:00-00:00",
    "name": "agent",
    "links": [],
    "alias": "agent",
```

```
        "description": "The agent management extension."
    }
}
```

Example 14.8. Get extension details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extension xmlns="http://openstack.org/quantum/api/v2.0"
           xmlns:quantum="http://openstack.org/quantum/api/v2.0"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <updated>2013-02-03T10:00:00-00:00</updated>
  <name>agent</name>
  <links quantum:type="list"/>
  <alias>agent</alias>
  <description>The agent management extension.</description>
</extension>
```

This operation does not return a response body.

14.2. Quotas extension (quotas)

Lists, shows information for, updates, and resets quotas.

Method	URI	Description
GET	/v2.0/quotas	Lists quotas for tenants who have non-default quota values.
GET	/v2.0/quotas/{tenant_id}	Shows 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.

14.2.1. List quotas

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)

14.2.1.1. Request

This operation does not accept a request body.

14.2.1.2. Response

Example 14.9. List quotas: JSON response

```
{
    "quotas": [
        {
            "subnet": 10,
            "ikepolicy": -1,
            "subnetpool": -1,
            "network": 15,
            "ipsec_site_connection": -1,
            "floatingip": 50,
            "tenant_id": "b7445f221cda4f4a8ac7db6b218b1339",
            "ipsecpolicy": -1,
            "security_group_rule": 100,
            "vpnservice": -1,
            "security_group": 10,
            "router": 10,
            "port": 30
        },
        {
            "subnet": 10,
            "ikepolicy": -1,
            "subnetpool": -1,
            "network": 5,
            "ipsec_site_connection": -1,
            "floatingip": 50,
            "tenant_id": "666a45fe39fe4e67bd3e542e8fd5087d",
            "ipsecpolicy": -1,
            "security_group_rule": 100,
            "vpnservice": -1,
            "security_group": 10,
            "router": 10,
            "port": 30
        }
    ]
}
```

14.2.2. Show quota

Method	URI	Description
GET	/v2.0/quotas/{tenant_id}	Shows quotas for a tenant.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

14.2.2.1. Request

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

Name	Type	Description
{tenant_id}	Uuid	The tenant ID.

This operation does not accept a request body.

14.2.2.2. Response

Example 14.10. Show quota: JSON response

```
{
    "quota": {
        "subnet": 10,
        "router": 10,
        "port": 50,
        "network": 10,
        "floatingip": 50,
        "subnetpool": -1,
        "security_group_rule": 100,
        "security_group": 1
    }
}
```

14.2.3. Update quota

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)

14.2.3.1. Request

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

Name	Type	Description
{tenant_id}	Uuid	The tenant ID.

Example 14.11. Update quota: JSON request

```
{
    "quota": {
        "subnet": 40,
        "network": 10,
        "health_monitor": 10,
        "subnetpool": -1,
        "security_group_rule": 100,
        "vip": 10,
        "member": -1,
        "floatingip": 50,
        "security_group": 10,
        "router": 50,
        "rbac_policy": -1,
        "port": 50,
        "pool": 10
    }
}
```

14.2.3.2. Response

Example 14.12. Update quota: JSON response

```
{
    "quota": {
        "subnet": 40,
        "network": 10,
        "health_monitor": 10,
        "subnetpool": -1,
        "security_group_rule": 100,
        "vip": 10,
        "member": -1,
        "floatingip": 50,
        "security_group": 10,
        "router": 50,
        "rbac_policy": -1,
        "port": 50,
        "pool": 10
    }
}
```

```
        "pool": 10
    }
}
```

14.2.4. Reset quota

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)

14.2.4.1. Request

This table shows the URI parameters for the reset quota request:

Name	Type	Description
{tenant_id}	Uuid	The tenant ID.

This operation does not accept a request body.

14.3. Networks provider extended attributes (networks)

Lists, creates, shows information for, updates, and deletes networks.

Method	URI	Description
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.

14.3.1. 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)

14.3.1.1. Request

This operation does not accept a request body.

14.3.1.2. Response

Example 14.13. List networks: JSON response

```
{
  "network": {
    "status": "ACTIVE",
    "subnets": [
      "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
    ],
    "name": "private-network",
    "router:external": false,
    "admin_state_up": true,
    "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
    "mtu": 0,
    "shared": true,
    "port_security_enabled": true,
    "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22"
  }
}
```

Example 14.14. List networks: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<network xmlns="http://openstack.org/quantum/api/v2.0"
          xmlns:provider="http://docs.openstack.org/ext/provider/api/v1.0"
          xmlns:quantum="http://openstack.org/quantum/api/v2.0"
          xmlns:router="http://docs.openstack.org/ext/neutron/router/api/v1.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <status>ACTIVE</status>
  <subnets>
    <subnet>54d6f61d-db07-451c-9ab3-b9609b6b6f0b</subnet>
  </subnets>
  <name>private-network</name>
  <admin_state_up quantum:type="bool">True</admin_state_up>
  <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
  <shared quantum:type="bool">True</shared>
  <id>d32019d3-bc6e-4319-9c1d-6722fc136a22</id>
</network>
```

This operation does not return a response body.

14.3.2. Create network

Method	URI	Description
POST	/v2.0/networks	Creates a network.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

14.3.2.1. Request

Example 14.15. Create network: JSON request

```
{
    "network": {
        "name": "sample_network",
        "admin_state_up": true
    }
}
```

Example 14.16. Create network: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<network>
    <name>sample_network2</name>
</network>
```

This operation does not accept a request body.

14.3.2.2. Response

Example 14.17. Create network: JSON response

```
{
    "network": {
        "status": "ACTIVE",
        "subnets": [],
        "name": "net1",
        "admin_state_up": true,
        "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
        "router:external": false,
        "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"  
    }  
}
```

Example 14.18. Create network: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<network xmlns="http://openstack.org/quantum/api/v2.0"  
         xmlns:provider="http://docs.openstack.org/ext/provider/api/v1.0"  
         xmlns:quantum="http://openstack.org/quantum/api/v2.0"  
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
    <status>ACTIVE</status>  
    <subnets quantum:type="list"/>  
    <name>sample_network2</name>  
    <provider:physical_network xsi:nil="true"/>  
    <admin_state_up quantum:type="bool">True</admin_state_up>  
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>  
    <provider:network_type>local</provider:network_type>  
    <shared quantum:type="bool">False</shared>  
    <id>c220b026-ece1-4ead-873f-83537f4c9f92</id>  
    <provider:segmentation_id xsi:nil="true"/>  
</network>
```

This operation does not return a response body.

14.3.3. 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)

14.3.3.1. Request

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

Name	Type	Description
{network_id}	UUID	The UUID for the network of interest to you.

This operation does not accept a request body.

14.3.3.2. Response

Example 14.19. 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",
        "mtu": 0,
        "shared": true,
        "port_security_enabled": true,
        "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22"
    }
}
```

Example 14.20. Show network details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<network xmlns="http://openstack.org/quantum/api/v2.0"
          xmlns:provider="http://docs.openstack.org/ext/provider/api/v1.0"
          xmlns:quantum="http://openstack.org/quantum/api/v2.0"
          xmlns:router="http://docs.openstack.org/ext/neutron/router/api/v1.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <status>ACTIVE</status>
    <subnets>
        <subnet>54d6f61d-db07-451c-9ab3-b9609b6b6f0b</subnet>
    </subnets>
    <name>private-network</name>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
    <shared quantum:type="bool">True</shared>
```

```
<id>d32019d3-bc6e-4319-9c1d-6722fc136a22</id>
</network>
```

This operation does not return a response body.

14.3.4. 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)

14.3.4.1. Request

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

Name	Type	Description
{network_id}	UUID	The UUID for the network of interest to you.

Example 14.21. Update network: JSON request

```
{
    "network": {
        "name": "sample_network_5_updated"
    }
}
```

Example 14.22. Update network: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<network xmlns="http://openstack.org/quantum/api/v2.0"
          xmlns:provider="http://docs.openstack.org/ext/provider/api/v1.0"
          xmlns:quantum="http://openstack.org/quantum/api/v2.0"
          xmlns:router="http://docs.openstack.org/ext/quantum/router/api/v1.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <name>sample-network-4-updated</name>
</network>
```

This operation does not accept a request body.

14.3.4.2. Response

Example 14.23. 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
    }
}
```

Example 14.24. Update network: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<network xmlns="http://openstack.org/quantum/api/v2.0"
          xmlns:provider="http://docs.openstack.org/ext/provider/api/v1.0"
          xmlns:quantum="http://openstack.org/quantum/api/v2.0"
          xmlns:router="http://docs.openstack.org/ext/neutron/router/api/v1.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <status>ACTIVE</status>
    <subnets quantum:type="list"/>
    <name>sample-network-4-updated</name>
    <provider:physical_network xsi:nil="true"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
    <provider:network_type>local</provider:network_type>
    <router:external quantum:type="bool">False</router:external>
    <shared quantum:type="bool">False</shared>
    <id>af374017-c9ae-4a1d-b799-ab73111476e2</id>
    <provider:segmentation_id xsi:nil="true"/>
</network>
```

This operation does not return a response body.

14.3.5. 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)

14.3.5.1. Request

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

Name	Type	Description
{network_id}	UUID	The UUID for the network of interest to you.

This operation does not accept a request body.

14.4. 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 <code>segments</code> 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.

14.4.1. 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)

14.4.1.1. Request

This operation does not accept a request body.

14.4.1.2. Response

Example 14.25. 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-34f7cf39c1b"
            ],
            "name": "private",
            "provider:physical_network": null,
            "router:external": true,
            "admin_state_up": true,
            "tenant_id": "26a7980765d0414dbcf1fc1f88cdb7e6e",
            "provider:network_type": "local",
        }
    ]
}
```

```
        "shared": true,
        "id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",
        "provider:segmentation_id": null
    }
]
}
```

14.4.2. 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)

14.4.2.1. Request

Example 14.26. 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
  }
}
```

14.4.2.2. Response

Example 14.27. 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"
      }
    ]
  }
}
```

```
        "provider:physical_network": "8bab8453-1bc9-45af-8c70-
f83aa9b50453",
        "provider:network_type": "stt"
    },
],
"shared": false,
"id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c"
}
}
```

14.4.3. 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)

14.4.3.1. 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 for the network of interest to you.

This operation does not accept a request body.

14.4.3.2. Response

Example 14.28. 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"
  }
}
```

14.5. 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	/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.

14.5.1. 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)

14.5.1.1. Request

This operation does not accept a request body.

14.5.1.2. Response

Example 14.29. 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"
        }
    ]
}
```

14.5.2. 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)

14.5.2.1. Request

Example 14.30. Create VLAN-transparent network: JSON request

```
{  
    "network": {  
        "name": "net1",  
        "admin_state_up": true,  
        "vlan_transparent": true  
    }  
}
```

14.5.2.2. Response

Example 14.31. 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"  
    }  
}
```

14.5.3. 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)

14.5.3.1. Request

This table shows the URI parameters for the show vlan-transparent network details request:

Name	Type	Description
{network_id}	UUID	The UUID for the network of interest to you.

This operation does not accept a request body.

14.5.3.2. Response

Example 14.32. 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"
    }
}
```

14.6. Ports binding extended attributes (ports)

Lists, creates, shows information for, and updates 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.

14.6.1. 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)

14.6.1.1. Request

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

Name	Type	Description
status <i>(Optional)</i>	String	The port status. Value is ACTIVE or DOWN.
display_name <i>(Optional)</i>	String	The port name.
admin_state <i>(Optional)</i>	Bool	The administrative state of the router, which is up (true) or down (false).
network_id <i>(Optional)</i>	Uuid	The UUID of the attached network.
tenant_id <i>(Optional)</i>	Uuid	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 <i>(Optional)</i>	String	The UUID of the entity that uses this port. For example, a DHCP agent.
mac_address <i>(Optional)</i>	String	The MAC address of the port.
port_id <i>(Optional)</i>	Uuid	The UUID of the port.
security_groups <i>(Optional)</i>	Uuid	The UUIDs of any attached security groups.
device_id <i>(Optional)</i>	Uuid	The UUID of the device that uses this port. For example, a virtual server.

14.6.1.2. Response

Example 14.33. List ports: JSON response

```
{
  "ports": [
    {
      "status": "ACTIVE",
      "binding:host_id": "devstack",
      "name": ""
    }
  ]
}
```

```
        "allowed_address_pairs": [],
        "admin_state_up": true,
        "network_id": "70c1db1f-b701-45bd-96e0-a313ee3430b3",
        "tenant_id": "",
        "extra_dhcp_opts": [],
        "binding:vif_details": {
            "port_filter": true,
            "ovs_hybrid_plug": true
        },
        "binding:vif_type": "ovs",
        "device_owner": "network:router_gateway",
        "port_security_enabled": true,
        "mac_address": "fa:16:3e:58:42:ed",
        "binding:profile": {},
        "binding:vnic_type": "normal",
        "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",
        "binding:host_id": "devstack",
        "name": "",
        "allowed_address_pairs": [],
        "admin_state_up": true,
        "network_id": "f27aa545-cbdd-4907-b0c6-c9e8b039dcc2",
        "tenant_id": "d397de8a63f341818f198abb0966f6f3",
        "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": true,
        "mac_address": "fa:16:3e:bb:3c:e4",
        "binding:profile": {},
        "binding:vnic_type": "normal",
        "fixed_ips": [
            {
                "subnet_id": "288bf4a1-51ba-43b6-9d0a-520e9005db17",
                "ip_address": "10.0.0.1"
            }
        ],
        "id": "f71a6703-d6de-4be1-a91a-a570edel159",
        "security_groups": [],
        "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824"
    }
]
```

Example 14.34. List ports: XML response

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

```
<ports xmlns="http://openstack.org/quantum/api/v2.0"
       xmlns:binding="http://docs.openstack.org/ext/binding/api/v1.0"
       xmlns:quantum="http://openstack.org/quantum/api/v2.0"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <port>
    <status>ACTIVE</status>
    <binding:host_id>devstack</binding:host_id>
    <name/>
    <allowed_address_pairs quantum:type="list"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <network_id>70c1db1f-b701-45bd-96e0-a313ee3430b3</network_id>
    <tenant_id/>
    <extra_dhcp_opts quantum:type="list"/>
    <binding:vif_details>
      <port_filter quantum:type="bool">True</port_filter>
      <ovs_hybrid_plug quantum:type="bool">
        >True</ovs_hybrid_plug>
    </binding:vif_details>
    <binding:vif_type>ovs</binding:vif_type>
    <device_owner>network:router_gateway</device_owner>
    <mac_address>fa:16:3e:58:42:ed</mac_address>
    <binding:profile quantum:type="dict"/>
    <binding:vnic_type>normal</binding:vnic_type>
    <fixed_ips>
      <fixed_ip>
        <subnet_id>008ba151-0b8c-4a67-98b5-0d2b87666062</subnet_id>
        <ip_address>172.24.4.2</ip_address>
      </fixed_ip>
    </fixed_ips>
    <id>d80b1a3b-4fc1-49f3-952e-1e2ab7081d8b</id>
    <security_groups quantum:type="list"/>
    <device_id>9ae135f4-b6e0-4dad-9e91-3c223e385824</device_id>
  </port>
  <port>
    <status>ACTIVE</status>
    <binding:host_id>devstack</binding:host_id>
    <name/>
    <allowed_address_pairs quantum:type="list"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <network_id>f27aa545-cbdd-4907-b0c6-c9e8b039dcc2</network_id>
    <tenant_id>d397de8a63f341818f198abb0966f6f3</tenant_id>
    <extra_dhcp_opts quantum:type="list"/>
    <binding:vif_details>
      <port_filter quantum:type="bool">True</port_filter>
      <ovs_hybrid_plug quantum:type="bool">
        >True</ovs_hybrid_plug>
    </binding:vif_details>
    <binding:vif_type>ovs</binding:vif_type>
    <device_owner>network:router_interface</device_owner>
    <mac_address>fa:16:3e:bb:3c:e4</mac_address>
    <binding:profile quantum:type="dict"/>
    <binding:vnic_type>normal</binding:vnic_type>
    <fixed_ips>
      <fixed_ip>
        <subnet_id>288bf4a1-51ba-43b6-9d0a-520e9005db17</subnet_id>
        <ip_address>10.0.0.1</ip_address>
      </fixed_ip>
    </fixed_ips>
    <id>f71a6703-d6de-4be1-a91a-a570edeld159</id>
    <security_groups quantum:type="list"/>
```

```
<device_id>9ae135f4-b6e0-4dad-9e91-3c223e385824</device_id>
</port>
</ports>
```

This operation does not return a response body.

14.6.2. 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)

14.6.2.1. Request

Example 14.35. Create port: JSON request

```
{
  "port": {
    "network_id": "ee2d3158-3e80-4fb3-ba87-c99f515d85e7",
    "admin_state_up": true
  }
}
```

14.6.2.2. Response

Example 14.36. 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": ""
  }
}
```

Example 14.37. Create port: XML response

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

```
<port xmlns="http://openstack.org/quantum/api/v2.0"
      xmlns:binding="http://docs.openstack.org/ext/binding/api/v1.0"
      xmlns:quantum="http://openstack.org/quantum/api/v2.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <status>DOWN</status>
    <binding:host_id/>
    <name>test_port_1</name>
    <allowed_address_pairs quantum:type="list"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <network_id>a87cc70a-3e15-4acf-8205-9b711a3531b7</network_id>
    <tenant_id>d6700c0c9ffa4f1cb322cd4a1f3906fa</tenant_id>
    <binding:vif_details quantum:type="dict"/>
    <binding:vnic_type>normal</binding:vnic_type>
    <binding:vif_type>unbound</binding:vif_type>
    <device_owner/>
    <mac_address>fa:16:3e:09:e3:47</mac_address>
    <binding:profile quantum:type="dict"/>
    <fixed_ips>
      <fixed_ip>
        <subnet_id>a0304c3a-4f08-4c43-88af-d796509c97d2</subnet_id>
        <ip_address>10.0.0.4</ip_address>
      </fixed_ip>
    </fixed_ips>
    <id>8021790b-4bfd-46ab-bcc7-0ef2f73bff43</id>
    <security_groups>
      <security_group>f0ac4394-7e4a-4409-9701-ba8be283dbc3</security_group>
    </security_groups>
    <device_id/>
  </port>
```

This operation does not return a response body.

14.6.3. Show port details

Method	URI	Description
GET	/v2.0/ports/{port_id}	Shows details for a port.

Normal response codes: 200

14.6.3.1. Request

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

Name	Type	Description
{port_id}	UUID	The UUID for the port of interest to you.

This operation does not accept a request body.

14.6.3.2. Response

Example 14.38. 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"
  }
}
```

Example 14.39. Show port details: XML response

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

```
<port xmlns="http://openstack.org/quantum/api/v2.0"
      xmlns:binding="http://docs.openstack.org/ext/binding/api/v1.0"
      xmlns:quantum="http://openstack.org/quantum/api/v2.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <status>ACTIVE</status>
    <binding:host_id>devstack</binding:host_id>
    <name/>
    <allowed_address_pairs quantum:type="list"/>
    <admin_state_up quantum:type="bool">True</admin_state_up>
    <network_id>a87cc70a-3e15-4acf-8205-9b711a3531b7</network_id>
    <tenant_id>7e02058126cc4950b75f9970368ba177</tenant_id>
    <extra_dhcp_opts quantum:type="list"/>
    <binding:vif_details>
        <port_filter quantum:type="bool">True</port_filter>
        <ovs_hybrid_plug quantum:type="bool">True</ovs_hybrid_plug>
    </binding:vif_details>
    <binding:vif_type>ovs</binding:vif_type>
    <device_owner>network:router_interface</device_owner>
    <mac_address>fa:16:3e:23:fd:d7</mac_address>
    <binding:profile quantum:type="dict"/>
    <binding:vnic_type>normal</binding:vnic_type>
    <fixed_ips>
        <fixed_ip>
            <subnet_id>a0304c3a-4f08-4c43-88af-d796509c97d2</subnet_id>
            <ip_address>10.0.0.1</ip_address>
        </fixed_ip>
    </fixed_ips>
    <id>46d4bfb9-b26e-41f3-bd2e-e6dcc1ccedb2</id>
    <security_groups quantum:type="list"/>
    <device_id>5e3898d7-11be-483e-9732-b2f5eccd2b2e</device_id>
  </port>
```

This operation does not return a response body.

14.6.4. 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)

14.6.4.1. Request

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

Name	Type	Description
{port_id}	UUID	The UUID for the port of interest to you.

Example 14.40. Update port: JSON request

```
{
  "port": {
    "network_id": "ee2d3158-3e80-4fb3-ba87-c99f515d85e7",
    "admin_state_up": true
  }
}
```

14.6.4.2. Response

Example 14.41. 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": ""
  }
}
```

```
    }  
}
```

Example 14.42. Update port: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<port xmlns="http://openstack.org/quantum/api/v2.0"  
      xmlns:binding="http://docs.openstack.org/ext/binding/api/v1.0"  
      xmlns:quantum="http://openstack.org/quantum/api/v2.0"  
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
  <status>DOWN</status>  
  <binding:host_id/>  
  <name>test_port_1</name>  
  <allowed_address_pairs quantum:type="list"/>  
  <admin_state_up quantum:type="bool">True</admin_state_up>  
  <network_id>a87cc70a-3e15-4acf-8205-9b711a3531b7</network_id>  
  <tenant_id>d6700c0c9ffa4f1cb322cd4a1f3906fa</tenant_id>  
  <binding:vif_details quantum:type="dict"/>  
  <binding:vnic_type>normal</binding:vnic_type>  
  <binding:vif_type>unbound</binding:vif_type>  
  <device_owner/>  
  <mac_address>fa:16:3e:09:e3:47</mac_address>  
  <binding:profile quantum:type="dict"/>  
  <fixed_ips>  
    <fixed_ip>  
      <subnet_id>a0304c3a-4f08-4c43-88af-d796509c97d2</subnet_id>  
      <ip_address>10.0.0.4</ip_address>  
    </fixed_ip>  
  </fixed_ips>  
  <id>8021790b-4bfd-46ab-bcc7-0ef2f73bff43</id>  
  <security_groups>  
    <security_group>f0ac4394-7e4a-4409-9701-ba8be283dbc3</security_group>  
  </security_groups>  
  <device_id/>  
</port>
```

This operation does not return a response body.

14.6.5. 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)

14.6.5.1. Request

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

Name	Type	Description
{port_id}	UUID	The UUID for the port of interest to you.

This operation does not accept a request body.

14.7. 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.

14.7.1. 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 unique ID for and the rules that are associated with each security group.

Normal response codes: 200

Error response codes: unauthorized (401)

14.7.1.1. Request

Example 14.43. List security groups: JSON request

```
GET /v2.0/security-groups
Accept: application/json
```

This operation does not accept a request body.

14.7.1.2. Response

Example 14.44. 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"
                }
            ]
        }
    ]
}
```

```
        "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"
}
]
```

14.7.2. 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)

14.7.2.1. Request

Example 14.45. Create security group: JSON request

```
{
    "security_group": {
        "name": "new-webservers",
        "description": "security group for webservers"
    }
}
```

14.7.2.2. Response

Example 14.46. 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-91e9-68885cff6ael",
                "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": "e4f50856753b4dc6af5fa6b9b6c550"
    }
],
"tenant_id": "e4f50856753b4dc6af5fa6b9b6c550"
}
```

14.7.3. 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, ID, and security group rules that are associated with the security group and tenant.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

14.7.3.1. Request

This table shows the URI parameters for the show security group request:

Name	Type	Description
{security_group_id}	Uuid	The unique identifier of the security group.

This table shows the query parameters for the show security group request:

Name	Type	Description
verbose	Bool <i>(Optional)</i>	Show detailed information.
fields	String <i>(Optional)</i>	The fields to be returned by server.

Example 14.47. 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.

14.7.3.2. Response

Example 14.48. 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,
        "tenant_id": "3c0e45ff-adaf-4124-b083-bf390e5482ff"
      }
    ]
  }
}
```

```
        "protocol": null,
        "remote_group_id": null,
        "remote_ip_prefix": null,
        "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "tenant_id": "e4f50856753b4dc6af5fa6b9b6c550"
    },
    {
        "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": "e4f50856753b4dc6af5fa6b9b6c550"
    },
    {
        "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": "e4f50856753b4dc6af5fa6b9b6c550"
    },
    {
        "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": "e4f50856753b4dc6af5fa6b9b6c550"
    }
],
"tenant_id": "e4f50856753b4dc6af5fa6b9b6c550"
}
```

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

14.7.4.1. Request

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

Name	Type	Description
{security_group_id}	Uuid	The unique identifier of the security group.

Example 14.49. Update security group: JSON request

```
{
  "security_group": {
    "name": "mysecgroup",
    "description": "my security group"
  }
}
```

14.7.4.2. Response

Example 14.50. Update security group: JSON response

```
{
  "security_group": {
    "rules": [],
    "tenant_id": "a52cdb9cc7854a39a23d3af73a40899e",
    "id": "01fbade5-b664-42f6-83ae-4e214f4263fa",
    "name": "mysecgroup",
    "description": "my security group"
  }
}
```

14.7.5. 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)

14.7.5.1. Request

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

Name	Type	Description
{security_group_id}	Uuid	The unique identifier of the security group.

Example 14.51. Delete security group: JSON request

```
DELETE /v2.0/security-groups/e470bd9c-4869-459b-a561-cb3377efae59
Content-Type: application/json
Accept: application/json
```

This operation does not accept a request body.

14.7.5.2. Response

Example 14.52. Delete security group: JSON response

```
status: 204
```

This operation does not return a response body.

14.8. 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/{rules-security-groups-id}	Shows detailed information for a security group rule.
DELETE	/v2.0/security-group-rules/{rules-security-groups-id}	Deletes a rule from an OpenStack Networking security group.

14.8.1. 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 unique ID for each security group rule.

Normal response codes: 200

Error response codes: unauthorized (401)

14.8.1.1. Request

Example 14.53. List security group rules: JSON request

```
GET /v2.0/security-group-rules/
Accept: application/json
```

This operation does not accept a request body.

14.8.1.2. Response

Example 14.54. 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"
    }
]
```

14.8.2. 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)

14.8.2.1. Request

Example 14.55. 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"
    }
}
```

14.8.2.2. Response

Example 14.56. 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"
    }
}
```

14.8.3. Show security group rule

Method	URI	Description
GET	/v2.0/security-group-rules/{rules-security-groups-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)

14.8.3.1. Request

This table shows the URI parameters for the show security group rule request:

Name	Type	Description
{rules-security-groups-id}	Uuid	The unique identifier of the security group rule.

Example 14.57. 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.

14.8.3.2. Response

Example 14.58. 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"
  }
}
```

14.8.4. Delete security group rule

Method	URI	Description
DELETE	/v2.0/security-group-rules/{rules-security-groups-id}	Deletes a rule from an OpenStack Networking security group.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404)

14.8.4.1. Request

This table shows the URI parameters for the delete security group rule request:

Name	Type	Description
{rules-security-groups-id}	Uuid	The unique identifier of the security group rule.

Example 14.59. 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.

14.8.4.2. Response

Example 14.60. Delete security group rule: JSON response

```
status: 204
```

This operation does not return a response body.

14.9. Layer-3 networking

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 is mapped to a port that is attached to an internal network.

Method	URI	Description
GET	/v2.0/routers	Lists logical routers that are accessible to the tenant who submits the request.
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	Removes 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.

14.9.1. List routers

Method	URI	Description
GET	/v2.0/routers	Lists logical routers that are accessible to the tenant who submits the request.

Default policy settings return only those routers that are owned by the tenant who submits the request, unless an admin 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)

14.9.1.1. Request

This operation does not accept a request body.

14.9.1.2. Response

Example 14.61. 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",
                "external_fixed_ips": [
                    {
                        "subnet_id": "255.255.255.0",
                        "ip": "192.168.10.2"
                    }
                ]
            },
            "name": "router1",
            "admin_state_up": true,
            "tenant_id": "33a40233088643acb6ff6eb0ebea679",
            "routes": [
                {
                    "status": "ACTIVE",
                    "external_gateway_info": {
                        "network_id": "3c5bcddd-6af9-4e6b-9c3e-c153e521cab8",
                        "external_fixed_ips": [
                            {
                                "subnet_id": "255.255.255.0",
                                "ip": "192.168.10.2"
                            }
                        ]
                    },
                    "name": "router1",
                    "admin_state_up": true,
                    "tenant_id": "33a40233088643acb6ff6eb0ebea679",
                    "routes": []
                }
            ]
        }
    ]
}
```

```
        "routes": [],
        "id": "a9254bdb-2613-4a13-ac4c-adc581fba50d"
    }
}
```

14.9.2. Create router

Method	URI	Description
POST	/v2.0/routers	Creates a logical router.

This operation creates a new logical router. When it is created, a logical router does not have any internal interface; it is not associated to 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 extended field `router:external` set to `true`. To specify an external gateway, the identifier 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)

14.9.2.1. Request

Example 14.62. 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
  }
}
```

14.9.2.2. Response

Example 14.63. 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",
        "routes": [],
        "id": "8604a0de-7f6b-409a-a47c-a1cc7bc77b2e"
    }
}
```

14.9.3. 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)

14.9.3.1. 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.

14.9.3.2. Response

Example 14.64. Show router details: JSON response

```
{
    "router": {
        "status": "ACTIVE",
        "external_gateway_info": {
            "network_id": "85d76829-6415-48ff-9c63-5c5ca8c61ac6",
            "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"
    }
}
```

14.9.4. 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)

14.9.4.1. Request

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

Name	Type	Description
{router_id}	UUID	The UUID of the router.

Example 14.65. 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"
        }
      ]
    }
  }
}
```

14.9.4.2. Response

Example 14.66. Update router: JSON response

```
{
  "router": {
    "status": "ACTIVE",
    "external_gateway_info": {
      "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b",
      "external_fixed_ips": [
        {

```

```
        "subnet_id": "255.255.255.0",
        "ip": "192.168.10.2"
    },
],
},
"name": "another_router",
"admin_state_up": true,
"tenant_id": "6b96ff0cb17a4b859e1e575d221683d3",
"routes": [],
"id": "8604a0de-7f6b-409a-a47c-a1cc7bc77b2e"
}
}
```

14.9.5. 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)

14.9.5.1. 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.

14.9.6. 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 a subnet ID or port ID in the request body:

- Subnet ID. The gateway IP address for the subnet is used to create the router interface.
- Port ID. 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 ID, on the router. If a port with the same network ID 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 ID.

When you use the `port-create` command to add a port and then call `router-interface-add` with this port ID, 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 ID as network ID of IPv6 subnets on any other ports.

If you specify both IDs, this operation returns the `Bad Request (400)` response code.

If the port is already in use, this operation returns the `Conflict (409)` response code.

This operation returns a port ID that is either:

- The same ID that is passed in the request body.
- The ID of a port that this operation creates to attach the specified subnet to the router.

After you run this operation, the operation sets:

- The device ID of this port to the router ID.
- The `device_owner` attribute to `network:router_interface`.

Normal response codes: 200

Error response codes: `badRequest (400)`, `unauthorized (401)`, `itemNotFound (404)`, `conflict (409)`

14.9.6.1. 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 14.67. Add interface to router: JSON request

```
{  
    "subnet_id": "a2f1f29d-571b-4533-907f-5803ab96ead1"  
}
```

14.9.6.2. Response**Example 14.68. Add interface to router: JSON response**

```
{  
    "subnet_id": "a2f1f29d-571b-4533-907f-5803ab96ead1",  
    "tenant_id": "6ba032e4730d42e2ad928f430f5da33e",  
    "port_id": "3a44f4e5-1694-493a-a1fb-393881c673a4",  
    "id": "b0294d7e-7da4-4202-9882-2ab1de9dabc0"  
}
```

14.9.7. Remove interface from router

Method	URI	Description
PUT	/v2.0/routers/{router_id}/remove_router_interface	Removes an internal interface from a logical router.

This operation removes an internal router interface, which detaches a subnet from the router. If this subnet ID is the last subnet on the port, this operation deletes the port itself. You must specify either a subnet ID or port ID in the request body; the operation uses this value to identify which router interface to remove.

You can also specify both a subnet ID and port ID. If you specify both IDs, the subnet ID must correspond to the subnet ID of the first IP address on the port specified by the port ID. 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 removes 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)`

14.9.7.1. Request

This table shows the URI parameters for the remove interface from router request:

Name	Type	Description
{router_id}	UUID	The UUID of the router.

Example 14.69. Remove interface from router: JSON request

```
{
    "subnet_id": "a2f1f29d-571b-4533-907f-5803ab96ead1"
}
```

14.9.7.2. Response

Example 14.70. Remove 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"
```

}

14.9.8. 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)

14.9.8.1. Request

This operation does not accept a request body.

14.9.8.2. Response

Example 14.71. List floating IPs: 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-clef-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"
        }
    ]
}
```

14.9.9. 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 ID attribute in the request body. If you do not specify a port ID 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 specific 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 ID of the network on which you want to create the floating IP. Alternatively, you can create a floating IP on a specific 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 ID 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)

14.9.9.1. Request

Example 14.72. Create floating IP: JSON request

```
{  
    "floatingip": {  
        "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",  
        "port_id": "ce705c24-clef-408a-bda3-7bbd946164ab"  
    }  
}
```

14.9.9.2. Response

Example 14.73. 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-clef-408a-bda3-7bbd946164ab",  
        "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",  
        "status": "ACTIVE",  
        "tenant_id": "4969c491a3c74ee4af974e6d800c62de"  
    }  
}
```

14.9.10. 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)

14.9.10.1. 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.

This operation does not accept a request body.

14.9.10.2. Response

Example 14.74. Show floating IP details: JSON response

```
{
  "floatingip": {
    "floating_network_id": "376da547-b977-4cf8-9cba-275c80deb57",
    "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-clef-408a-bda3-7bbd946164ab",
    "id": "2f245a7b-796b-4f26-9cf9-9e82d248fd7"
  }
}
```

14.9.11. 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)

14.9.11.1. 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.

Example 14.75. Update floating IP (associate port): JSON

```
{
  "floatingip": {
    "port_id": "fc861431-0e6c-4842-a0ed-e2363f9bc3a8"
  }
}
```

Example 14.76. Update floating IP (disassociate port): JSON

```
{
  "floatingip": {
    "port_id": null
  }
}
```

14.9.11.2. Response

Example 14.77. Update floating IP (associate port): JSON

```
{
  "floatingip": {
    "floating_network_id": "376da547-b977-4cfe-9cba-275c80deb57",
```

```
        "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 14.78. Update floating IP (disassociate port): JSON

```
{
    "floatingip": {
        "floating_network_id": "376da547-b977-4cfe-9cba-275c80deb5f57",
        "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"
    }
}
```

14.9.12. 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)

14.9.12.1. 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.

This operation does not accept a request body.

14.10. Metering labels and rules

Creates, modifies, and deletes OpenStack Layer3 metering labels and rules.

Method	URI	Description
GET	/v2.0/metering/metering-labels	Lists all I3 metering labels that belong to the tenant.
POST	/v2.0/metering/metering-labels	Creates an I3 metering label.
GET	/v2.0/metering/metering-labels/{metering_label_id}	Shows information for a metering label.
DELETE	/v2.0/metering/metering-labels/{metering_label_id}	Deletes an I3 metering label.
GET	/v2.0/metering/metering-label-rules	Lists a summary of all I3 metering label rules that belong to the tenant.
POST	/v2.0/metering/metering-label-rules	Creates an I3 metering label rule.
GET	/v2.0/metering/metering-label-rules/{metering_label_rule_id}	Shows detailed information for a metering label rule.
DELETE	/v2.0/metering/metering-label-rules/{metering_label_rule_id}	Deletes a I3 metering label rule.

14.10.1. List metering labels

Method	URI	Description
GET	/v2.0/metering/metering-labels	Lists all I3 metering labels that belong to the tenant.

The list shows the unique ID for each metering label.

Normal response codes: 200

Error response codes: unauthorized (401)

14.10.1.1. Request

Example 14.79. 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.

14.10.1.2. Response

Example 14.80. 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
        }
    ]
}
```

14.10.2. Create metering label

Method	URI	Description
POST	/v2.0/metering/metering-labels	Creates an I3 metering label.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), forbidden (403)

14.10.2.1. Request

Example 14.81. Create metering label: JSON request

```
{  
    "metering_label": {  
        "name": "label1",  
        "description": "description of label1"  
    }  
}
```

14.10.2.2. Response

Example 14.82. Create metering label: JSON response

```
{  
    "metering_label": {  
        "tenant_id": "45345b0ee1ea477fac0f541b2cb79cd4",  
        "description": "description of label1",  
        "name": "label1",  
        "id": "bc91b832-8465-40a7-a5d8-ba87de442266",  
        "shared": false  
    }  
}
```

14.10.3. Show metering label

Method	URI	Description
GET	/v2.0/metering/metering-labels/{metering_label_id}	Shows information for a metering label.

The response body shows the description, name, ID.

Normal response codes: 200

Error response codes: unauthorized (401), itemNotFound (404)

14.10.3.1. Request

This table shows the URI parameters for the show metering label request:

Name	Type	Description
{metering_label_id}	Uuid	The unique identifier of the metering label.

Example 14.83. Show metering label: 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.

14.10.3.2. Response

Example 14.84. Show metering label: JSON response

```
{
    "metering_label": {
        "tenant_id": "45345b0ee1ea477fac0f541b2cb79cd4",
        "description": "label1 description",
        "name": "label1",
        "id": "a6700594-5b7a-4105-8bfe-723b346ce866",
        "shared": false
    }
}
```

14.10.4. Delete metering label

Method	URI	Description
DELETE	/v2.0/metering/metering-labels/{metering_label_id}	Deletes an I3 metering label.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404)

14.10.4.1. Request

This table shows the URI parameters for the delete metering label request:

Name	Type	Description
{metering_label_id}	Uuid	The unique identifier of the metering label.

Example 14.85. 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.

14.10.4.2. Response

Example 14.86. Delete metering label: JSON response

```
status: 204
```

This operation does not return a response body.

14.10.5. List metering label rules

Method	URI	Description
GET	/v2.0/metering/metering-label-rules	Lists a summary of all I3 metering label rules that belong to the tenant.

The list shows the unique ID for each metering label rule.

Normal response codes: 200

Error response codes: unauthorized (401)

14.10.5.1. Request

Example 14.87. 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.

14.10.5.2. Response

Example 14.88. 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
        }
    ]
}
```

14.10.6. Create metering label rule

Method	URI	Description
POST	/v2.0/metering/metering-label-rules	Creates an I3 metering label rule.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), conflict (409)

14.10.6.1. Request

Example 14.89. 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"  
    }  
}
```

14.10.6.2. Response

Example 14.90. 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  
    }  
}
```

14.10.7. Show metering label rule

Method	URI	Description
GET	/v2.0/metering/metering-label-rules/{metering-label-rule-id}	Shows detailed information for a metering label rule.

The response body shows the following information for each metering label rule:

- direction. Either ingress or egress.
- excluded. Either True or False.
- The ID 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)

14.10.7.1. Request

This table shows the URI parameters for the show metering label rule request:

Name	Type	Description
{metering-label-rule-id}	Uuid	The unique identifier of metering label rule.

Example 14.91. Show metering label rule: 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.

14.10.7.2. Response

Example 14.92. Show metering label rule: 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
    }
}
```


14.10.8. Delete metering label rule

Method	URI	Description
DELETE	/v2.0/metering/metering-label-rules/{metering-label-rule-id}	Deletes a I3 metering label rule.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404)

14.10.8.1. Request

This table shows the URI parameters for the delete metering label rule request:

Name	Type	Description
{metering-label-rule-id}	Uuid	The unique identifier of metering label rule.

Example 14.93. 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.

14.10.8.2. Response

Example 14.94. Delete metering label rule: JSON response

```
status: 204
```

This operation does not return a response body.

14.11. Firewall-as-a-Service (FWaaS) 2.0 (CURRENT)

The FWaaS extension enables you 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 is associated with 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 be different from the tenant who created or the tenants that use the firewall policy.
- **firewall_rule**. A collection of attributes like ports and IP addresses that define match criteria and action, such as allow or deny, that must be taken on the matched data traffic.

Method	URI	Description
GET	/v2.0/fw/firewalls	List 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}	Removes a firewall.

14.11.1. List firewalls

Method	URI	Description
GET	/v2.0/fw/firewalls	List firewalls.

Lists all firewalls. The list might be empty.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403)

14.11.1.1. Request

This operation does not accept a request body.

14.11.1.2. Response

Example 14.95. 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",  
            "tenant_id": "45977fa2dbd7482098dd68d0d8970117"  
        }  
    ]  
}
```

14.11.2. Create firewall

Method	URI	Description
POST	/v2.0/fw/firewalls	Creates a firewall.

Creates a firewall object. The firewall must be associated with a firewall policy.

Example:

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

14.11.2.1. Request

Example 14.96. Create firewall: JSON request

```
{  
    "firewall": {  
        "admin_state_up": true,  
        "firewall_policy_id": "c69933c1-b472-44f9-8226-30dc4ffd454c"  
    }  
}
```

14.11.2.2. Response

Example 14.97. 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",  
        "tenant_id": "45977fa2dbd7482098dd68d0d8970117"  
    }  
}
```

14.11.3. Show firewall details

Method	URI	Description
GET	/v2.0/fw/firewalls/{firewall_id}	Shows details for a firewall.

Shows the details for a firewall. If the user is not an administrative user and the firewall object does not belong to the user's tenant account, a 403 (Forbidden) error is returned.

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404)

14.11.3.1. Request

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

Name	Type	Description
{firewall_id}	UUID	The UUID for the firewall.

This operation does not accept a request body.

14.11.3.2. Response

Example 14.98. 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",
    "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
  }
}
```

14.11.4. Update firewall

Method	URI	Description
PUT	/v2.0/fw/firewalls/{firewall_id}	Updates a firewall.

Updates the attributes for 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)

14.11.4.1. Request

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

Name	Type	Description
{firewall_id}	UUID	The UUID for the firewall.

Example 14.99. Update firewall: JSON request

```
{
  "firewall": {
    "admin_state_up": "false"
  }
}
```

14.11.4.2. Response

Example 14.100. 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",
    "tenant_id": "45977fa2dbd7482098dd68d0d8970117"
  }
}
```

14.11.5. Remove firewall

Method	URI	Description
DELETE	/v2.0/fw/firewalls/{firewall_id}	Removes a firewall.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

14.11.5.1. Request

This table shows the URI parameters for the remove firewall request:

Name	Type	Description
{firewall_id}	UUID	The UUID for the firewall.

Example 14.101. Remove firewall: JSON request

```
{
    "firewall": {
        "id": "3b0ef8f4-82c7-44d4-a4fb-6177f9a21977"
    }
}
```

14.12. Load-Balancer-as-a-Service (LBaaS) 1.0 (STABLE)

The LBaaS version 1.0 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 virtual IP addresses (VIPs), pools, members of a pool, health monitors associated with a pool, and view status of a resource.

Table 14.1. Load balancer statuses

Status	Description
ACTIVE	Resource is ready and active.
PENDING_CREATE	Resource is being created.
PENDING_UPDATE	Resource is being updated.
PENDING_DELETE	Resource is pending deletion.
INACTIVE	Resource was created but is not active.
ERROR	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.

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.

14.12.1. List VIPs

Method	URI	Description
GET	/v2.0/lb/vips	Lists VIPs.

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

14.12.1.1. Request

This operation does not accept a request body.

14.12.1.2. Response

Example 14.102. List VIPs: JSON response

```
{
    "vips": [
        {
            "status": "ACTIVE",
            "protocol": "HTTP",
            "description": "",
            "admin_state_up": true,
            "subnet_id": "8032909d-47a1-4715-90af-5153ffe39861",
            "tenant_id": "83657cfcdfe44cd5920adaf26c48ceea",
            "connection_limit": 1000,
            "pool_id": "72741b06-df4d-4715-b142-276b6bce75ab",
            "session_persistence": {
                "cookie_name": "MyAppCookie",
                "type": "APP_COOKIE"
            },
            "address": "10.0.0.10",
            "protocol_port": 80,
            "port_id": "b5a743d6-056b-468b-862d-fb13a9aa694e",
            "id": "4ec89087-d057-4e2c-911f-60a3b47ee304",
            "name": "my-vip"
        }
    ]
}
```

14.12.2. 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: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

14.12.2.1. Request

Example 14.103. Create a load balancer VIP: JSON request

```
{
    "vip": {
        "protocol": "HTTP",
        "name": "NewVip",
        "admin_state_up": true,
        "subnet_id": "8032909d-47a1-4715-90af-5153ffe39861",
        "pool_id": "61b1f87a-7a21-4ad3-9dda-7f81d249944f",
        "protocol_port": "80"
    }
}
```

14.12.2.2. Response

Example 14.104. Create a load balancer VIP: JSON response

```
{
    "vip": {
        "status": "PENDING_CREATE",
        "protocol": "HTTP",
        "description": "",
        "admin_state_up": true,
        "subnet_id": "8032909d-47a1-4715-90af-5153ffe39861",
        "tenant_id": "83657cfcdfe44cd5920adaf26c48ceea",
        "connection_limit": -1,
        "pool_id": "61b1f87a-7a21-4ad3-9dda-7f81d249944f",
        "address": "10.0.0.11",
        "protocol_port": 80,
        "port_id": "f7e6fe6a-b8b5-43a8-8215-73456b32e0f5",
        "id": "c987d2be-9a3c-4ac9-a046-e8716b1350e2",
        "name": "NewVip"
    }
}
```

14.12.3. 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)

14.12.3.1. 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.

14.12.3.2. Response

Example 14.105. Show VIP details: JSON response

```
{
    "vip": {
        "status": "ACTIVE",
        "protocol": "HTTP",
        "description": "",
        "admin_state_up": true,
        "subnet_id": "8032909d-47a1-4715-90af-5153ffe39861",
        "tenant_id": "83657cfcdfe44cd5920adaf26c48ceea",
        "connection_limit": 1000,
        "pool_id": "72741b06-df4d-4715-b142-276b6bce75ab",
        "session_persistence": {
            "cookie_name": "MyAppCookie",
            "type": "APP_COOKIE"
        },
        "address": "10.0.0.10",
        "protocol_port": 80,
        "port_id": "b5a743d6-056b-468b-862d-fb13a9aa694e",
        "id": "4ec89087-d057-4e2c-911f-60a3b47ee304",
        "name": "my-vip"
    }
}
```

14.12.4. 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)

14.12.4.1. Request

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

Name	Type	Description
{vip_id}	UUID	The UUID for the VIP.

Example 14.106. Update VIP: JSON request

```
{
    "vip": {
        "connection_limit": "1000"
    }
}
```

14.12.4.2. Response

Example 14.107. Update VIP: JSON response

```
{
    "vip": {
        "status": "PENDING_UPDATE",
        "protocol": "HTTP",
        "description": "",
        "admin_state_up": true,
        "subnet_id": "8032909d-47a1-4715-90af-5153ffe39861",
        "tenant_id": "83657cfcdfe44cd5920adaf26c48ceea",
        "connection_limit": 1000,
        "pool_id": "61b1f87a-7a21-4ad3-9dda-7f81d249944f",
        "address": "10.0.0.11",
        "protocol_port": 80,
        "port_id": "f7e6fe6a-b8b5-43a8-8215-73456b32e0f5",
        "id": "c987d2be-9a3c-4ac9-a046-e8716b1350e2",
        "name": "NewVip"
    }
}
```

14.12.5. 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)

14.12.5.1. 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.

14.12.6. 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)

14.12.6.1. Request

This operation does not accept a request body.

14.12.6.2. Response

Example 14.108. List health monitors: JSON response

```
{
  "health_monitors": [
    {
      "admin_state_up":true,
      "tenant_id":"83657cfcdfe44cd5920adaf26c48ceea",
      "delay":10,
      "max_retries":1,
      "timeout":1,
      "type":"PING",
      "id":"466c8345-28d8-4f84-a246-e04380b0461d"
    },
    {
      "admin_state_up":true,
      "tenant_id":"83657cfcdfe44cd5920adaf26c48ceea",
      "delay":5,
      "expected_codes":"200",
      "max_retries":2,
      "http_method":"GET",
      "timeout":2,
      "url_path":"/",
      "type":"HTTP",
      "id":"5d4b5228-33b0-4e60-b225-9b727c1a20e7"
    }
  ]
}
```

14.12.7. 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)

14.12.7.1. Request

Example 14.109. Create a load balancer health monitor: JSON request

```
{
  "healthmonitor": {
    "admin_state_up": true,
    "delay": "1",
    "expected_codes": "200,201,202",
    "http_method": "GET",
    "max_retries": 5,
    "pool_id": "74aa2010-a59f-4d35-a436-60a6da882819",
    "timeout": 1,
    "type": "HTTP",
    "url_path": "/index.html"
  }
}
```

14.12.7.2. Response

Example 14.110. Create a load balancer health monitor: JSON response

```
{
  "healthmonitor": {
    "admin_state_up": true,
    "delay": 1,
    "expected_codes": "200,201,202",
    "http_method": "GET",
    "id": "0a9ac99d-0a09-4b18-8499-a0796850279a",
    "max_retries": 5,
    "pools": [
      {
        "id": "74aa2010-a59f-4d35-a436-60a6da882819"
      }
    ],
    "tenant_id": "6f3584d5754048a18e30685362b88411",
    "timeout": 1,
    "type": "HTTP",
    "url_path": "/index.html"
  }
}
```

14.12.8. 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)

14.12.8.1. 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.

14.12.8.2. Response

Example 14.111. Show health monitor details: JSON response

```
{
  "healthmonitor": {
    "admin_state_up": true,
    "delay": 1,
    "expected_codes": "200,201,202",
    "http_method": "GET",
    "id": "0a9ac99d-0a09-4b18-8499-a0796850279a",
    "max_retries": 5,
    "pools": [
      {
        "id": "74aa2010-a59f-4d35-a436-60a6da882819"
      }
    ],
    "tenant_id": "6f3584d5754048a18e30685362b88411",
    "timeout": 1,
    "type": "HTTP",
    "url_path": "/index.html"
  }
}
```

14.12.9. 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)

14.12.9.1. 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 14.112. Update health monitor: JSON request

```
{
  "health_monitor": {
    "delay": "3"
  }
}
```

14.12.9.2. Response

Example 14.113. Update health monitor: JSON response

```
{
  "health_monitor": {
    "admin_state_up": true,
    "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
    "delay": 5,
    "max_retries": 5,
    "http_method": "GET",
    "timeout": 1,
    "pools": [
      {
        "status": "PENDING_CREATE",
        "status_description": null,
        "pool_id": "6e55751f-6ad4-4e53-b8d4-02e442cd21df"
      }
    ],
    "type": "PING",
    "id": "b05e44b5-81f9-4551-b474-711a722698f7"
  }
}
```

14.12.10. 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)

14.12.10.1. 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.

14.12.11. 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)

14.12.11.1. Request

This operation does not accept a request body.

14.12.11.2. Response

Example 14.114. List pools: JSON response

```
{
  "pools": [
    {
      "status": "ACTIVE",
      "lb_method": "ROUND_ROBIN",
      "protocol": "HTTP",
      "description": "",
      "health_monitors": [
        "466c8345-28d8-4f84-a246-e04380b0461d",
        "5d4b5228-33b0-4e60-b225-9b727c1a20e7"
      ],
      "subnet_id": "8032909d-47a1-4715-90af-5153ffe39861",
      "tenant_id": "83657cfcdfe44cd5920adaf26c48ceea",
      "admin_state_up": true,
      "name": "app_pool",
      "members": [
        "701b531b-111a-4f21-ad85-4795b7b12af6",
        "beb53b4d-230b-4abd-8118-575b8fa006ef"
      ],
      "id": "72741b06-df4d-4715-b142-276b6bce75ab",
      "vip_id": "4ec89087-d057-4e2c-911f-60a3b47ee304"
    }
  ]
}
```

14.12.12. 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)

14.12.12.1. Request

Example 14.115. Create a load balancer pool: JSON request

```
{
  "pool": {
    "admin_state_up": true,
    "description": "simple pool",
    "lb_algorithm": "ROUND_ROBIN",
    "listener_id": "39de4d56-d663-46e5-85a1-5b9d5fa17829",
    "name": "pool1",
    "protocol": "HTTP",
    "session_persistence": {
      "cookie_name": "my_cookie",
      "type": "APP_COOKIE"
    }
  }
}
```

14.12.12.2. Response

Example 14.116. Create a load balancer pool: JSON response

```
{
  "pool": {
    "admin_state_up": true,
    "description": "simple pool",
    "healthmonitor_id": null,
    "id": "12ff63af-4127-4074-a251-bcb2ecc53ebe",
    "lb_algorithm": "ROUND_ROBIN",
    "listeners": [
      {
        "id": "39de4d56-d663-46e5-85a1-5b9d5fa17829"
      }
    ],
    "members": [],
    "name": "pool1",
    "protocol": "HTTP",
    "session_persistence": {
      "cookie_name": "my_cookie",
      "type": "APP_COOKIE"
    },
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c"
  }
}
```

14.12.13. 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)

14.12.13.1. 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.

14.12.13.2. Response

Example 14.117. Show pool details: JSON response

```
{
  "pool": {
    "admin_state_up": true,
    "description": "simple pool",
    "healthmonitor_id": null,
    "id": "4c0a0a5f-cf8f-44b7-b912-957daa8ce5e5",
    "lb_algorithm": "ROUND_ROBIN",
    "listeners": [
      {
        "id": "35cb8516-1173-4035-8dae-0dae3453f37f"
      }
    ],
    "members": [],
    "name": "pool1",
    "protocol": "HTTP",
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c"
  }
}
```

14.12.14. 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)

14.12.14.1. Request

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

Name	Type	Description
{pool_id}	UUID	The UUID for the pool.

Example 14.118. Update pool: JSON request

```
{
  "pool": {
    "name": "SuperPool"
  }
}
```

14.12.14.2. Response

Example 14.119. Update pool: JSON response

```
{
  "pool": {
    "status": "PENDING_UPDATE",
    "lb_method": "ROUND_ROBIN",
    "protocol": "TCP",
    "description": "",
    "health_monitors": [
      ,
      "subnet_id": "8032909d-47a1-4715-90af-5153ffe39861",
      "tenant_id": "83657cfcdfe44cd5920adaf26c48ceea",
      "admin_state_up": true,
      "name": "SuperPool",
      "members": [
        ,
        "id": "61b1f87a-7a21-4ad3-9dda-7f81d249944f",
        "vip_id": null
      ]
    }
  }
}
```

14.12.15. 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)

14.12.15.1. 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.

14.12.16. 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)

14.12.16.1. 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 14.120. Associate health monitor with pool: JSON request

```
{
  "health_monitor": {
    "id": "b624decf-d5d3-4c66-9a3d-f047e7786181"
  }
}
```

14.12.16.2. Response

Example 14.121. Associate health monitor with pool: JSON response

```
{
  "health_monitor": {
  }
}
```

14.12.17. 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)

14.12.17.1. 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.

14.12.18. 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)

14.12.18.1. Request

This operation does not accept a request body.

14.12.18.2. Response

Example 14.122. List members: JSON response

```
{
  "members": [
    {
      "status": "ACTIVE",
      "weight": 1,
      "admin_state_up": true,
      "tenant_id": "83657cfcdfe44cd5920adaf26c48ceea",
      "pool_id": "72741b06-df4d-4715-b142-276b6bce75ab",
      "address": "10.0.0.4",
      "protocol_port": 80,
      "id": "701b531b-111a-4f21-ad85-4795b7b12af6"
    },
    {
      "status": "ACTIVE",
      "weight": 1,
      "admin_state_up": true,
      "tenant_id": "83657cfcdfe44cd5920adaf26c48ceea",
      "pool_id": "72741b06-df4d-4715-b142-276b6bce75ab",
      "address": "10.0.0.3",
      "protocol_port": 80,
      "id": "beb53b4d-230b-4abd-8118-575b8fa006ef"
    }
  ]
}
```

14.12.19. 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)

14.12.19.1. Request

Example 14.123. Create a load balancer member: JSON request

```
{  
    "member": {  
        "address": "10.0.0.8",  
        "admin_state_up": true,  
        "protocol_port": "80",  
        "subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",  
        "weight": "1"  
    }  
}
```

14.12.19.2. Response

Example 14.124. Create a load balancer member: JSON response

```
{  
    "member": {  
        "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  
    }  
}
```

14.12.20. 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)

14.12.20.1. 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.

14.12.20.2. Response

Example 14.125. Show member details: JSON response

```
{
  "member": {
    "address": "10.0.0.8",
    "admin_state_up": true,
    "id": "9a7aff27-fd41-4ec1-ba4c-3eb92c629313",
    "protocol_port": 80,
    "pool_id": "a5a8839d-1ac3-41f9-9aae-f375fa4da50a",
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
    "weight": 1
  }
}
```

14.12.21. 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)

14.12.21.1. Request

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

Name	Type	Description
{member_id}	UUID	The UUID for the member.

Example 14.126. Update member: JSON request

```
{
  "member": {
    "admin_state_up": false
  }
}
```

14.12.21.2. Response

Example 14.127. Update member: JSON response

```
{
  "member": {
    "status": "PENDING_UPDATE",
    "protocol_port": 8080,
    "weight": 1,
    "admin_state_up": false,
    "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
    "pool_id": "7803631d-f181-4500-b3a2-1b68ba2a75fd",
    "address": "10.0.0.5",
    "status_description": null,
    "id": "48a471ea-64f1-4eb6-9be7-dae6bbe40a0f"
  }
}
```

14.12.22. 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)

14.12.22.1. 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.

14.13. Load-Balancer-as-a-Service (LBaaS) 2.0 (EXPERIMENTAL)

The 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 associated with a pool and view status of a resource.

Table 14.2. Load balancer statuses

Status	Description
ACTIVE	Resource is ready and active.
PENDING_CREATE	Resource is being created.
PENDING_UPDATE	Resource is being updated.
PENDING_DELETE	Resource is pending deletion.
INACTIVE	Resource was created but is not active.
ERROR	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 load balancers.
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/listeners	Lists 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 of 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/{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.

14.13.1. List load balancers

Method	URI	Description
GET	/v2.0/lbaas/loadbalancers	Lists load balancers.

Lists all load balancers that are associated with your tenant account.

This operation returns a list, which might be empty. Each element in the list is a load balancer that can contain the following attributes:

- id
- tenant_id
- name
- description
- vip_subnet_id
- vip_address
- admin_state_up
- listeners
- provisioning_status
- operating_status

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

14.13.1.1. Request

This operation does not accept a request body.

14.13.1.2. Response

Example 14.128. List load balancers: JSON response

```
{  
    "loadbalancers": [  
        {  
            "id": "3b98602c-3cf8-4f91-bfa4-c3a11c9e7fe0",  
            "name": "Example LB",  
            "description": "A very simple example load balancer.",  
            "tenant_id": "783b31af-6635-48b2-a807-091d9973e3a9",  
            "admin_state_up": true,  
            "status": "ACTIVE"  
        },  
        {  
        }  
    ]  
}
```

```
        "id": "c617c538-daa5-4ead-be88-59521d8745a7",
        "name": "Example LB",
        "description": "A very simple example load balancer.",
        "tenant_id": "783b31af-6635-48b2-a807-091d9973e3a9",
        "admin_state_up": true,
        "status": "ACTIVE"
    }
]
}
```

14.13.2. Create load balancer

Method	URI	Description
POST	/v2.0/lbaas/loadbalancers	Creates a load balancer.

This operation provisions a new load balancer based on the configuration defined in the request object. After the request is validated and progress has started on the provisioning process, a response object is returned. The object contains a unique identifier and the status of provisioning the load balancer.

The `provisioning_status` of the load balancer in the response can have one of the following values: ACTIVE, PENDING_CREATE, or ERROR.

If the status is PENDING_CREATE, the caller can view the progress of the provisioning operation by performing a **GET** on `/lbaas/loadbalancers/loadbalancer_id`. When the status of the load balancer changes to ACTIVE, the load balancer was successfully provisioned and is operational for traffic handling.

If the request cannot be fulfilled due to insufficient or invalid 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.

You can configure all documented features of the load balancer at creation time by specifying the additional elements or attributes in the request.

Users with an administrative role can create load balancers on behalf of other tenants by specifying a `tenant_id` attribute different than their own.

Example: Create a load balancer

- `tenant_id`. Only required if the caller has an administrative role and wants 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 are authorized by the policy, 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 request cannot be fulfilled 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.

Users with an administrative role can create load balancers on behalf of other tenants by specifying a `tenant_id` attribute that is different than their own.

A user can supply a `vip_address` field if she owns the subnet on which the load balancer's VIP will be created. If a `vip_address` is omitted from the payload, the LBaaS service allocates a VIP address from the subnet of the load balancer VIP.

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)

14.13.2.1. Request

Example 14.129. 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  
    }  
}
```

14.13.2.2. Response

Example 14.130. 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"  
    }  
}
```

14.13.3. 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 identified by `loadbalancer_id`. If the user is not an administrative user and the load balancer object does not belong to her tenant account, the service returns the HTTP `Forbidden` (403) response code.

If this operation succeeds, it returns a load balancer element that can contain the following attributes:

- `id`
- `tenant_id`
- `name`
- `description`
- `vip_subnet_id`
- `vip_address`
- `admin_state_up`
- `listeners`
- `provisioning_status`
- `operating_status`

Example: Show load balancer details

Normal response codes: 200

Error response codes: unauthorized (401), forbidden (403), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

14.13.3.1. Request

This operation does not accept a request body.

14.13.3.2. Response

Example 14.131. Show load balancer details: JSON response

```
{
  "loadbalancer":{
    "id":"8992a43f-83af-4b49-9af8-c2bfbd82d7d7",
    "name":"Example LB",
    "description":"A very simple example load balancer.",
```

```
        "vip_address": "1.2.3.4",
        "vip_subnet_id": "SUBNET_ID",
        "tenant_id": "7725fe12-1c14-4f45-ba8e-44bf01763578",
        "admin_state_up": true,
        "status": "ACTIVE"
    }
}
```

14.13.4. Update load balancer

Method	URI	Description
PUT	/v2.0/lbaas/loadbalancers/{loadbalancer_id}	Updates a load balancer.

Upon successful validation of the request, the service returns the Accepted (202) response code. A caller should check that the load balancer provisioning_status has changed to ACTIVE to confirm that the update has taken effect. If the load balancer provisioning_status is PENDING_UPDATE, the caller can poll the load balancer object by using a GET operation to wait for the changes to be applied.

The update operation enables you to change one or more of the following load balancer attributes:

- name
- description
- admin_state_up

This operation returns the updated load balancer object. The provisioning_status value can be ACTIVE, PENDING_UPDATE, or ERROR.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

14.13.4.1. Request

Example 14.132. Update load balancer: JSON request

```
{
  "loadbalancer": {
    "admin_state_up": false,
    "description": "simple lb2",
    "name": "loadbalancer2"
  }
}
```

14.13.4.2. Response

Example 14.133. 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"
    }
}
```

14.13.5. 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.

Any and all configuration data is immediately purged and cannot be recovered.

Example: Delete a load balancer

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

14.13.5.1. Request

This operation does not accept a request body.

14.13.6. List listeners

Method	URI	Description
GET	/v2.0/lbaas/listeners	Lists listeners.

This operation lists all listeners that are associated with your tenant account.

This operation returns a list, which might be empty. Each list element is a listener that can contain the following attributes:

- `id`
- `tenant_id`
- `name`
- `description`
- `protocol`
- `protocol_port`
- `connection_limit`
- `default_pool_id`
- `admin_state_up`
- `loadbalancers`
- `default_tls_container_ref`
- `sni_container_refs`

Example: List listeners

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

14.13.6.1. Request

This operation does not accept a request body.

14.13.6.2. Response

Example 14.134. List listeners: JSON response

```
{  
    "listeners": [  
        {  
            "admin_state_up": true,  
            "connection_limit": 100,  
            "default_pool_id": "pool-1",  
            "description": "Listener for https traffic",  
            "idle_timeout": 5000,  
            "loadbalancers": [  
                {"admin_state_up": true,  
                 "connection_limit": 100,  
                 "idle_timeout": 5000,  
                 "name": "lb-1",  
                 "pool_id": "pool-1"},  
                {"admin_state_up": true,  
                 "connection_limit": 100,  
                 "idle_timeout": 5000,  
                 "name": "lb-2",  
                 "pool_id": "pool-1"}],  
            "name": "https-listener",  
            "protocol": "https",  
            "protocol_port": 443,  
            "sni_container_refs": ["sni-1"]  
        }]  
}
```

```
    "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": "3e4d8bec50a845fc09e03a4375c691d",
    "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"
    ]
}
}
```

14.13.7. Create listener

Method	URI	Description
POST	/v2.0/lbaas/listeners	Creates a listener.

This operation provisions a new listener based on the configuration defined 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`. Required only if the caller has an administrative role and wants to create a listener for another tenant.
- `loadbalancer_id`. The load balancer on which this listener is provisioned. A tenant can only create listeners on load balancers authorized by policy. 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 that are used 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 request cannot be fulfilled due to insufficient or invalid 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.

You can configure all documented features of the listener at creation time by specifying the additional elements or attributes in the request.

Users with an administrative role can create listeners on behalf of other tenants by specifying a `tenant_id` attribute different than their own.

A listener cannot be updated if the load balancer that it is attempting to be attached to does not have a `provisioning_status` of ACTIVE.

Example: Create a listener

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

14.13.7.1. Request

Example 14.135. 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"  
        ]  
    }  
}
```

14.13.7.2. Response

Example 14.136. 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",  
        "updated_at": "2015-11-17T14:45:29Z",  
        "version": 1  
    }  
}
```

```
        "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"  
        ]  
    }  
}
```

14.13.8. Show listener details

Method	URI	Description
GET	/v2.0/lbaas/listeners/{listener_id}	Shows details for a listener.

This operation returns a listener object identified by `listener_id`. If the user is not an administrative user and the listener object does not belong to her tenant account, the call returns the HTTP `Forbidden` (403) response code.

If this operation succeeds, it returns a `listener` element that can contain the following attributes:

- `id`
- `tenant_id`
- `name`
- `description`
- `protocol`
- `protocol_port`
- `connection_limit`
- `default_pool_id`
- `admin_state_up`
- `loadbalancers`
- `default_tls_container_ref`
- `sni_container_refs`

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)

14.13.8.1. Request

This operation does not accept a request body.

14.13.8.2. Response

Example 14.137. 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"
        ]
    }
}
```

14.13.9. 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.

The update operation enables the caller to change one or more of the following listener attributes:

- name
- description
- admin_state_up
- connection_limit
- default_tls_container_ref
- sni_container_refs

Example: Update a listener

Note: You cannot update the `listener_id`, `tenant_id`, `loadbalancer_id`, `load-balancers`, `default_pool_id`, `protocol`, and `protocol_port` listener 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 a `provisioning_status` of ACTIVE.

Normal response codes: 200

Error response codes: Internal-server-error (500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

14.13.9.1. Request

Example 14.138. 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/b36c20d0-18e9-42ce-88fd-82a35977ee8e"
    ]
  }
}
```

```
        "https://barbican.endpoint/containers/
c36c20d0-18e9-42ce-88fd-82a35977ee8e"
    ]
}
}
```

14.13.9.2. Response

Example 14.139. 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"
    ]
  }
}
```

14.13.10. 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. Any and all configuration data is immediately purged and cannot be recovered.

You cannot delete a listener if the load balancer to which it is attached does not have a provisioning_status of ACTIVE.

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)

14.13.10.1. Request

This operation does not accept a request body.

14.13.11. List pools

Method	URI	Description
GET	/v2.0/lbaas/pools	Lists all pools that are associated with your tenant account.

This operation returns a response body that contains a list, which might be empty. Each pool element in the list can contain the following attributes:

- `id`
- `tenant_id`
- `name`
- `description`
- `protocol`
- `lb_algorithm`
- `session_persistence`
- `admin_state_up`
- `listeners`
- `members`
- `health_monitor_id`

Example: List pools

Normal response codes: 200

Error response codes: unauthorized (401), Internal-server-error (500), serviceUnavailable (503)

14.13.11.1. Request

This operation does not accept a request body.

14.13.11.2. Response

Example 14.140. List pools: JSON response

```
{
  "pools": [
    {
      "admin_state_up": true,
      "description": "simple pool",
      "healthmonitor_id": null,
      "id": "4c0a0a5f-cf8f-44b7-b912-957daa8ce5e5",
      "lb_algorithm": "ROUND_ROBIN",
      "members": [
        {
          "admin_state_up": true,
          "id": "4c0a0a5f-cf8f-44b7-b912-957daa8ce5e5",
          "loadbalancer_id": "4c0a0a5f-cf8f-44b7-b912-957daa8ce5e5",
          "member_id": "4c0a0a5f-cf8f-44b7-b912-957daa8ce5e5",
          "port": 80
        }
      ],
      "name": "simple pool"
    }
  ]
}
```

```
    "listeners": [
        {
            "id": "35cb8516-1173-4035-8dae-0dae3453f37f"
        }
    ],
    "members": [],
    "name": "pool1",
    "protocol": "HTTP",
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c"
}
]
```

14.13.12. Create pool

Method	URI	Description
POST	/v2.0/lbaas/pools	Creates a pool.

This operation provisions a new pool based on the configuration defined in the request object. After the request is validated and progress has started on the provisioning process, a response object is returned. The object contains a unique identifier.

The caller of this operation must specify these pool attributes, at a minimum:

- `tenant_id`. Required only if the caller has an administrative role and wants 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 is used to distribute traffic to the pool members. This value, which must be supported, is dependent on the load-balancer provider.
- `listener_id`. The ID of the listener in which this pool becomes the default pool. Each listener can have 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 request cannot be fulfilled due to insufficient or invalid 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.

Users can configure all documented features at creation time by providing the additional elements or attributes in the request.

Users with an administrative role can create pools on behalf of other tenants by specifying a `tenant_id` attribute that is different than their own.

You cannot update a pool if the load balancer to which it is attempting to be attached does not have a `provisioning_status` of ACTIVE.

Example: Create a pool

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

14.13.12.1. Request

Example 14.141. Create pool: JSON request

```
{  
    "pool": {  
        "admin_state_up": true,  
        "description": "simple pool",  
        "lb_algorithm": "ROUND_ROBIN",  
        "listener_id": "39de4d56-d663-46e5-85a1-5b9d5fa17829",  
        "name": "pool1",  
        "protocol": "HTTP",  
        "session_persistence": {  
            "cookie_name": "my_cookie",  
            "type": "APP_COOKIE"  
        }  
    }  
}
```

14.13.12.2. Response

Example 14.142. Create pool: JSON response

```
{  
    "pool": {  
        "admin_state_up": true,  
        "description": "simple pool",  
        "healthmonitor_id": null,  
        "id": "12ff63af-4127-4074-a251-bcb2ecc53ebe",  
        "lb_algorithm": "ROUND_ROBIN",  
        "listeners": [  
            {  
                "id": "39de4d56-d663-46e5-85a1-5b9d5fa17829"  
            }  
        ],  
        "members": [],  
        "name": "pool1",  
        "protocol": "HTTP",  
        "session_persistence": {  
            "cookie_name": "my_cookie",  
            "type": "APP_COOKIE"  
        },  
        "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c"  
    }  
}
```

14.13.13. Show pool details

Method	URI	Description
GET	/v2.0/lbaas/pools/{pool_id}	Shows details for a pool.

This operation returns a pool object identified by `pool_id`. If the user is not an administrative user and the pool object does not belong to her tenant account, the call returns the HTTP Forbidden (403) response code.

If this operation succeeds, it returns a `pool` element that can contain the following attributes:

- `id`
- `tenant_id`
- `name`
- `description`
- `protocol`
- `lb_algorithm`
- `session_persistence`
- `admin_state_up`
- `listeners`
- `members`
- `health_monitor_id`

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)

14.13.13.1. Request

This operation does not accept a request body.

14.13.13.2. Response

Example 14.143. Show pool details: JSON response

```
{
    "pool": {
        "admin_state_up": true,
        "description": "simple pool",
        "id": "pool-12345678901234567890123456789012",
        "lb_algorithm": "ROUND_ROBIN",
        "members": [
            {
                "address": "192.168.1.100",
                "id": "member-12345678901234567890123456789012",
                "port": 80
            }
        ],
        "name": "my_pool",
        "protocol": "HTTP"
    }
}
```

```
"healthmonitor_id": null,  
"id": "4c0a0a5f-cf8f-44b7-b912-957daa8ce5e5",  
"lb_algorithm": "ROUND_ROBIN",  
"listeners": [  
    {  
        "id": "35cb8516-1173-4035-8dae-0dae3453f37f"  
    }  
,  
    "members": [],  
    "name": "pool1",  
    "protocol": "HTTP",  
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c"  
}  
}
```

14.13.14. 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.

The update operation enables the caller to change one or more of the following pool attributes:

- name
- description
- admin_state_up
- lb_algorithm
- session_persistence

Note: You cannot update the pool ID, 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 a provisioning_status of ACTIVE.

Example: Update a pool

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

14.13.14.1. Request

Example 14.144. Update pool: JSON request

```
{
  "pool": {
    "admin_state_up": false,
    "description": "pool two",
    "lb_algorithm": "LEAST_CONNECTIONS",
    "name": "pool2",
    "session_persistence": {
      "type": "HTTP_COOKIE"
    }
  }
}
```

14.13.14.2. Response

Example 14.145. Update pool: JSON response

```
{  
    "pool": {  
        "admin_state_up": false,  
        "description": "pool two",  
        "healthmonitor_id": null,  
        "id": "12ff63af-4127-4074-a251-bcb2ecc53ebe",  
        "lb_algorithm": "LEAST_CONNECTIONS",  
        "listeners": [  
            {  
                "id": "39de4d56-d663-46e5-85a1-5b9d5fa17829"  
            }  
        ],  
        "members": [],  
        "name": "pool2",  
        "protocol": "HTTP",  
        "session_persistence": {  
            "cookie_name": null,  
            "type": "HTTP_COOKIE"  
        },  
        "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c"  
    }  
}
```

14.13.15. 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. Any and all configuration data is immediately purged and cannot be recovered.

You cannot delete a pool if the load balancer to which it is attached does not have a provisioning_status of ACTIVE.

Example: Delete a pool

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

14.13.15.1. Request

This operation does not accept a request body.

14.13.16. 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`.

This operation returns a list, which might be empty. Each element in the list is a member that can contain the following attributes:

- `id`
- `tenant_id`
- `address`
- `protocol_port`
- `weight`
- `subnet_id`
- `admin_state_up`

Example: List pool members

Normal response codes: 200

Error response codes: unauthorized (401), serviceUnavailable (503), Internal-server-error (500)

14.13.16.1. Request

This operation does not accept a request body.

14.13.16.2. Response

Example 14.146. 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
    }
  ]
}
```

}

14.13.17. 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 new member and adds it to a pool based on the configuration defined in the request object. After the request is validated and progress has started on the provisioning process, a response object is returned. The object contains a unique identifier.

At a minimum, you must specify the following pool attributes:

- `tenant_id`. Only required if the caller has an administrative role and wants 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 is listening to receive 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 ID.

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.

Users with an administrative role can create members on behalf of other tenants by specifying a `tenant_id` attribute that is different than their own.

To update a member, the load balancer must have a `provisioning_status` of ACTIVE.

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

14.13.17.1. Request

Example 14.147. Add member to a pool: JSON request

```
{
  "member": {
    "address": "10.0.0.8",
    "admin_state_up": true,
```

```
        "protocol_port": "80",
        "subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
        "weight": "1"
    }
}
```

14.13.17.2. Response

Example 14.148. Add member to pool: JSON response

```
{
    "member": {
        "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
    }
}
```

14.13.18. 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 the user is not an administrative user and the pool or member object does not belong to her tenant account, the service returns the HTTP Forbidden (403) response code.

If this operation succeeds, it returns a pool element that can contain the following attributes:

- `id`
- `tenant_id`
- `address`
- `protocol_port`
- `weight`
- `subnet_id`
- `admin_state_up`

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)

14.13.18.1. Request

This operation does not accept a request body.

14.13.18.2. Response

Example 14.149. Show pool member details: JSON response

```
{
  "member": {
    "address": "10.0.0.8",
    "admin_state_up": true,
    "id": "9a7aff27-fd41-4ec1-ba4c-3eb92c629313",
    "protocol_port": 80,
    "pool_id": "a5a8839d-1ac3-41f9-9aae-f375fa4da50a",
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
    "weight": 1
  }
}
```

}

14.13.19. Update pool member

Method	URI	Description
PUT	/v2.0/lbaas/pools/{pool_id}/members/{member_id}	Updates attributes of a pool member.

Upon successful validation of the request, the service returns the **HTTP OK (200) response code**.

The update operation enables you to change one or more of these pool attributes:

- weight
- admin_state_up

Note: You cannot update the member ID, 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 a provisioning_status of ACTIVE.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

14.13.19.1. Request

Example 14.150. Update pool member: JSON request

```
{
  "member": {
    "admin_state_up": false,
    "weight": 5
  }
}
```

14.13.19.2. Response

Example 14.151. Update pool member: JSON response

```
{
  "member": {
    "address": "10.0.0.8",
    "admin_state_up": false,
    "id": "9a7aff27-fd41-4ec1-ba4c-3eb92c629313",
    "protocol_port": 80,
    "subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
    "weight": 5
  }
}
```

14.13.20. 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.

Any and all configuration data is immediately purged and cannot be recovered.

A member cannot be deleted if the attached load balancer does not have a provisioning_status of ACTIVE.

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)

14.13.20.1. Request

This operation does not accept a request body.

14.13.21. Create health monitor

Method	URI	Description
POST	/v2.0/lbaas/health_monitors	Creates a health monitor.

This operation provisions a new health monitor based on the configuration defined in the request object. After the request is validated and progress has started on the provisioning process, a response object is returned. The object contains a unique identifier.

The caller of this operation must specify these health monitor attributes, at a minimum:

- `tenant_id`. Only required if the caller has an administrative role and wants to create a health monitor for another tenant.
- `type`. The type of health monitor. Must be one of TCP, HTTP, HTTPS
- `delay`. The interval in seconds between health checks.
- `timeout`. The time in seconds that a health check times out.
- `max_retries`. Number of failed health checks before marked as OFFLINE.
- `pool_id`. The pool that this health monitor will 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 request cannot be fulfilled due to insufficient data or data that is not valid, an HTTP 400 (Bad Request) error response is returned with information regarding the nature of the failure in the response body. Failures in the validation process are non-recoverable and require the caller to correct the cause of the failure and **POST** 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.

Users with an administrative role can create health monitors on behalf of other tenants by specifying a `tenant_id` attribute different than their own.

To update a health monitor, the load balancer to which it is being attached must have an ACTIVE provisioning status.

Example: Create a health monitor

Normal response codes: 201

Error response codes: unauthorized (401), itemNotFound (404), conflict (409), overLimit (413), Internal-server-error (500), serviceUnavailable (503)

14.13.21.1. Request

Example 14.152. Create health monitor: JSON request

```
{  
    "healthmonitor": {  
        "admin_state_up": true,  
        "delay": "1",  
        "expected_codes": "200,201,202",  
        "http_method": "GET",  
        "max_retries": 5,  
        "pool_id": "74aa2010-a59f-4d35-a436-60a6da882819",  
        "timeout": 1,  
        "type": "HTTP",  
        "url_path": "/index.html"  
    }  
}
```

14.13.21.2. Response

Example 14.153. Create health monitor: JSON response

```
{  
    "healthmonitor": {  
        "admin_state_up": true,  
        "delay": 1,  
        "expected_codes": "200,201,202",  
        "http_method": "GET",  
        "id": "0a9ac99d-0a09-4b18-8499-a0796850279a",  
        "max_retries": 5,  
        "pools": [  
            {  
                "id": "74aa2010-a59f-4d35-a436-60a6da882819"  
            }  
        ],  
        "tenant_id": "6f3584d5754048a18e30685362b88411",  
        "timeout": 1,  
        "type": "HTTP",  
        "url_path": "/index.html"  
    }  
}
```

14.13.22. 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 identified by `health_monitor_id`. If the user is not an administrative user and the health monitor object does not belong to her tenant account, the service returns the HTTP Forbidden (403) response code.

If this operation succeeds, it returns a health monitor element that can contain the following attributes:

- `id`
- `tenant_id`
- `type`
- `delay`
- `timeout`
- `max_retries`
- `http_method`
- `url_path`
- `expected_codes`
- `admin_state_up`
- `pool_id`
- `pools`

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)

14.13.22.1. Request

This operation does not accept a request body.

14.13.22.2. Response

Example 14.154. Show health monitor details: JSON response

```
{
```

```
"healthmonitor": {
    "admin_state_up": true,
    "delay": 1,
    "expected_codes": "200,201,202",
    "http_method": "GET",
    "id": "0a9ac99d-0a09-4b18-8499-a0796850279a",
    "max_retries": 5,
    "pools": [
        {
            "id": "74aa2010-a59f-4d35-a436-60a6da882819"
        }
    ],
    "tenant_id": "6f3584d5754048a18e30685362b88411",
    "timeout": 1,
    "type": "HTTP",
    "url_path": "/index.html"
}
}
```

14.13.23. 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.

The update operation enables you to change one or more health monitor attributes:

- delay
- timeout
- max_retries
- http_method
- url_path
- expected_codes
- admin_state_up

Note: The health monitor ID, 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)

14.13.23.1. Request

Example 14.155. Update health monitor: JSON request

```
{
  "healthmonitor": {
    "admin_state_up": false,
    "delay": "2",
    "expected_codes": "200",
    "http_method": "POST",
    "max_retries": 2,
    "timeout": 2,
    "url_path": "/page.html"
  }
}
```

14.13.23.2. Response

Example 14.156. Update health monitor: JSON response

```
{
```

```
"healthmonitor": {
    "admin_state_up": false,
    "delay": 2,
    "expected_codes": "200",
    "http_method": "POST",
    "id": "0a9ac99d-0a09-4b18-8499-a0796850279a",
    "max_retries": 2,
    "pools": [
        {
            "id": "74aa2010-a59f-4d35-a436-60a6da882819"
        }
    ],
    "tenant_id": "6f3584d5754048a18e30685362b88411",
    "timeout": 2,
    "type": "HTTP",
    "url_path": "/page.html"
}
}
```

14.13.24. 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.

Any and all configuration data is immediately purged and cannot be recovered.

You cannot delete a health monitor if the attached load balancer does not have a provisioning_status value of ACTIVE.

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)

14.13.24.1. Request

This operation does not accept a request body.

14.14. Subnet pools extension (subnetpools)

Manages 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.

14.14.1. 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)

14.14.1.1. Request

This operation does not accept a request body.

14.14.1.2. Response

Example 14.157. 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,
            "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,
            "tenant_id": "9fadcee8aa7c40cdb2114fff7d569c08",
            "prefixes": [
                "10.10.0.0/21",
                "192.168.0.0/16"
            ],
            "ip_version": 4,
            "shared": false
        }
    ]
}
```

}

14.14.2. 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)

14.14.2.1. Request

Example 14.158. 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"
    }
}
```

14.14.2.2. Response

Example 14.159. 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,
        "tenant_id": "9fadcee8aa7c40cdb2114ffff7d569c08",
        "prefixes": [
            "10.10.0.0/21",
            "192.168.0.0/16"
        ],
        "ip_version": 4,
        "shared": false
    }
}
```

14.14.3. 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)

14.14.3.1. Request

This table shows the URI parameters for the show subnet pool request:

Name	Type	Description
{subnetpool_id}	UUID	The UUID for the subnet pool of interest to you.

This operation does not accept a request body.

14.14.3.2. Response

Example 14.160. 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,
        "tenant_id": "9fadcee8aa7c40cdb2114ffff7d569c08",
        "prefixes": [
            "2001:db8:0:2::/64",
            "2001:db8::/63"
        ],
        "ip_version": 6,
        "shared": false
    }
}
```

14.14.4. 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)

14.14.4.1. Request

This table shows the URI parameters for the update subnet pool request:

Name	Type	Description
{subnetpool_id}	UUID	The UUID for the subnet pool of interest to you.

Example 14.161. 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
    }
}
```

14.14.4.2. Response

Example 14.162. Update subnet pool: JSON response

```
{
    "subnetpool": {
        "name": "my-new-subnetpool-name",
        "default_quota": null,
        "tenant_id": "9fadcee8aa7c40cdb2114ffff7d569c08",
        "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
    }
}
```


14.14.5. 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)

14.14.5.1. Request

This table shows the URI parameters for the delete subnet pool request:

Name	Type	Description
{subnetpool_id}	UUID	The UUID for the subnet pool of interest to you.

This operation does not accept a request body.

14.15. Virtual-Private-Network-as-a-Service (VPNaas) 2.0 (CURRENT)

The 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.

14.15.1. 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)

14.15.1.1. Request

This operation does not accept a request body.

14.15.1.2. Response

Example 14.163. 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": ""
        }
    ]
}
```

14.15.2. 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.

Example:

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401)

14.15.2.1. Request

Example 14.164. Create VPN service: JSON request

```
{  
    "vpnservice": {  
        "subnet_id": null,  
        "router_id": "66e3b16c-8ce5-40fb-bb49-ab6d8dc3f2aa",  
        "name": "myservice",  
        "admin_state_up": true  
    }  
}
```

14.15.2.2. Response

Example 14.165. 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": ""  
    }  
}
```

14.15.3. 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)

14.15.3.1. Request

This table shows the URI parameters for the show vpn service details request:

Name	Type	Description
{service_id}	UUID	The UUID for the VPN service.

This operation does not accept a request body.

14.15.3.2. Response

Example 14.166. 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": ""
    }
}
```

14.15.4. 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)

14.15.4.1. Request

This table shows the URI parameters for the update vpn service request:

Name	Type	Description
{service_id}	UUID	The UUID for the VPN service.

Example 14.167. Update VPN service: JSON request

```
{
    "vpnservice": {
        "description": "Updated description"
    }
}
```

14.15.4.2. Response

Example 14.168. 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"
    }
}
```

14.15.5. 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)

14.15.5.1. Request

This table shows the URI parameters for the remove vpn service request:

Name	Type	Description
{service_id}	UUID	The UUID for the VPN service.

This operation does not accept a request body.

14.15.6. 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)

14.15.6.1. Request

This operation does not accept a request body.

14.15.6.2. Response

Example 14.169. 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": ""
        }
    ]
}
```

14.15.7. 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)

14.15.7.1. Request

Example 14.170. 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"
  }
}
```

14.15.7.2. Response

Example 14.171. 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": ""
  }
}
```

14.15.8. 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)

14.15.8.1. Request

This table shows the URI parameters for the show ike policy details request:

Name	Type	Description
{ikepolicy_id}	UUID	The UUID for the IKE policy.

This operation does not accept a request body.

14.15.8.2. Response

Example 14.172. 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": ""
  }
}
```

14.15.9. 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)

14.15.9.1. Request

This table shows the URI parameters for the update ike policy request:

Name	Type	Description
{ikepolicy_id}	UUID	The UUID for the IKE policy.

Example 14.173. Update IKE policy: JSON request

```
{
  "ikepolicy": {
    "encryption_algorithm": "aes-256"
  }
}
```

14.15.9.2. Response

Example 14.174. 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": ""
  }
}
```

14.15.10. Remove IKE policy

Method	URI	Description
DELETE	/v2.0/vpn/ikepoli-cies/{ikepolicy_id}	Removes an IKE policy.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

14.15.10.1. Request

This table shows the URI parameters for the remove ike policy request:

Name	Type	Description
{ikepolicy_id}	UUID	The UUID for the IKE policy.

This operation does not accept a request body.

14.15.11. 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)

14.15.11.1. Request

This operation does not accept a request body.

14.15.11.2. Response

Example 14.175. 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": ""
        }
    ]
}
```

14.15.12. Create IPSec policy

Method	URI	Description
POST	/v2.0/vpn/ipsecolicies	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)

14.15.12.1. Request

Example 14.176. 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
    }
  }
}
```

14.15.12.2. Response

Example 14.177. 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": ""
  }
}
```

14.15.13. 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)

14.15.13.1. Request

This table shows the URI parameters for the show ipsec policy request:

Name	Type	Description
{ipsecpolicy_id}	UUID	The UUID for the IPSec policy.

This operation does not accept a request body.

14.15.13.2. Response

Example 14.178. 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": ""
    }
}
```

14.15.14. Update IPSec policy

Method	URI	Description
PUT	/v2.0/vpn/ipsecolicies/{ipsecpolicy_id}	Updates policy settings in an IPSec policy.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

14.15.14.1. Request

This table shows the URI parameters for the update ipsec policy request:

Name	Type	Description
{ipsecpolicy_id}	UUID	The UUID for the IPSec policy.

Example 14.179. Update IPSec policy: JSON request

```
{
    "ipsecpolicy": {
        "pfs": "group14"
    }
}
```

14.15.14.2. Response

Example 14.180. 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": ""
    }
}
```

14.15.15. Remove IPSec policy

Method	URI	Description
DELETE	/v2.0/vpn/ipsecolicies/{ipsecpolicy_id}	Removes an IPSec policy.

Normal response codes: 204

Error response codes: unauthorized (401), itemNotFound (404), conflict (409)

14.15.15.1. Request

This table shows the URI parameters for the remove ipsec policy request:

Name	Type	Description
{ipsecpolicy_id}	UUID	The UUID for the IPSec policy.

This operation does not accept a request body.

14.15.16. 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)

14.15.16.1. Request

This operation does not accept a request body.

14.15.16.2. Response

Example 14.181. 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"
        }
    ]
}
```

14.15.17. 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)

14.15.17.1. Request

Example 14.182. Create VPN endpoint group: JSON request

```
{
  "endpoint_group": {
    "endpoints": [
      "10.2.0.0/24",
      "10.3.0.0/24"
    ],
    "type": "cidr",
    "name": "peers"
  }
}
```

14.15.17.2. Response

Example 14.183. 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"
  }
}
```

14.15.18. 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)

14.15.18.1. Request

This table shows the URI parameters for the show vpn endpoint group request:

Name	Type	Description
{endpoint_group_id}	UUID	The UUID for the VPN endpoint group.

This operation does not accept a request body.

14.15.18.2. Response

Example 14.184. 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"
    }
}
```

14.15.19. 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)

14.15.19.1. Request

This table shows the URI parameters for the update vpn endpoint group request:

Name	Type	Description
{endpoint_group_id}	UUID	The UUID for the VPN endpoint group.

Example 14.185. Update VPN endpoint group: JSON request

```
{
    "endpoint_group": {
        "description": "New description"
    }
}
```

14.15.19.2. Response

Example 14.186. 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"
    }
}
```

14.15.20. 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)

14.15.20.1. Request

This table shows the URI parameters for the remove vpn endpoint group request:

Name	Type	Description
{endpoint_group_id}	UUID	The UUID for the VPN endpoint group.

This operation does not accept a request body.

14.15.21. 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)

14.15.21.1. Request

This operation does not accept a request body.

14.15.21.2. Response

Example 14.187. 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-a7b8645fdd1",
            "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": ""
        }
    ]
}
```

14.15.22. 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)

14.15.22.1. Request

Example 14.188. 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-a7b8645fdd1",
    "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"
  }
}
```

14.15.22.2. Response

Example 14.189. 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-a7b8645fdd1",
    "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",
    "id": "10039663455a446d8ba2cbb058b0f578"
  }
}
```

```
        "peer_address": "172.24.4.233",
        "peer_id": "172.24.4.233",
        "id": "851f280f-5639-4ea3-81aa-e298525ab74b",
        "description": ""
    }
}
```

14.15.23. 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)

14.15.23.1. Request

This table shows the URI parameters for the show ipsec connection request:

Name	Type	Description
{connection_id}	UUID	The UUID for the IPSec site-to-site connection.

This operation does not accept a request body.

14.15.23.2. Response

Example 14.190. 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-a7b8645fdddf1",
        "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": ""
    }
}
```

14.15.24. 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)

14.15.24.1. Request

This table shows the URI parameters for the update ipsec connection request:

Name	Type	Description
{connection_id}	UUID	The UUID for the IPSec site-to-site connection.

Example 14.191. Update IPSec connection: JSON request

```
{
    "ipsec_site_connection": {
        "mtu": "2000"
    }
}
```

14.15.24.2. Response

Example 14.192. 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-a7b8645fdd1",
        "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"
    }
}
```

```
    }  
}
```

14.15.25. 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)

14.15.25.1. Request

This table shows the URI parameters for the remove ipsec connection request:

Name	Type	Description
{connection_id}	UUID	The UUID for the IPSec site-to-site connection.

This operation does not accept a request body.

14.16. Extra routes

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}	Configures extra routes on a router.

14.16.1. Update router

Method	URI	Description
PUT	/v2.0/routers/{router_id}	Configures 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 400 Bad Request error code.

When a validation error is detected, such as a format error of IP address or CIDR, the server responds with the 400 Bad Request error 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 409 Conflict error code.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), conflict (409)

14.16.1.1. Request

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

Name	Type	Description
{router_id}	UUID	The UUID for the router of interest to you.

Example 14.193. Update router: JSON request

```
{
  "router": {
    "routes": [
      {
        "nexthop": "10.1.0.10",
        "destination": "40.0.1.0/24"
      }
    ]
  }
}
```

14.16.1.2. Response

Example 14.194. Update router: JSON response

```
{
  "router": {
    "status": "ACTIVE",
    "external_gateway_info": {
      "network_id": "5c26e0bb-a9a9-429c-9703-5c417a221096",
      "external_fixed_ips": [
        {
          "subnet_id": "255.255.255.0",
          "ip": "192.168.10.2"
        }
      ]
    }
  }
}
```

```
        },
        "name": "router1",
        "admin_state_up": true,
        "tenant_id": "936fa220b2c24a87af51026439af7a3e",
        "routes": [
            {
                "nexthop": "10.1.0.10",
                "destination": "40.0.1.0/24"
            }
        ],
        "id": "babcb8173-46f6-4b6f-8b95-38c1683a4e22"
    }
}
```

15. 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](#).

Method	URI	Description
Discoverability		
GET	/info{?swiftinfo_sig, swiftinfo_expires}	Lists the activated capabilities for this version of the OpenStack Object Storage API.
Endpoints		
GET	/v1/endpoints	Lists endpoints for an object, account, or container.
Accounts		
GET	/v1/{account}{?limit,marker,end_marker,format,prefix,delim-iter}	Shows details for an account and lists containers, sorted by name, in the account.
POST	/v1/{account}	Creates, updates, or deletes account metadata.
HEAD	/v1/{account}	Shows metadata for an account.
Containers		
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.

Method	URI	Description
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.

15.1. 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.

15.1.1. 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

15.1.1.1. Request

This table shows the query parameters for the list activated capabilities request:

Name	Type	Description
swiftinfo_sig	Char <i>(Optional)</i>	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 <i>(Optional)</i>	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.

15.1.1.2. Response

Example 15.1. List activated capabilities: JSON response

```
{
    "swift": {
        "version": "1.11.0"
    },
    "staticweb": {},
    "tempurl": {}
}
```

15.2. Endpoints

If configured, lists endpoints for an account.

Method	URI	Description
GET	/v1/endpoints	Lists endpoints for an object, account, or container.

15.2.1. 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

15.2.1.1. Request

This operation does not accept a request body.

15.2.1.2. Response

Example 15.2. 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 15.3. 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"
  }
}
```

15.3. 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 15.1. 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,delim-iter}</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.

15.3.1. Show account details and list containers

Method	URI	Description
GET	/v1/{account}{?limit,marker,end_marker,format,prefix,delim-iter}	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: 200, 204

15.3.1.1. 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
	(Required)	
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.

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 (Optional)	For an integer value n , limits the number of results to n .
marker	String (Optional)	For a string value, x , returns container names that are greater than the marker value.
end_marker	String (Optional)	For a string value, x , returns container names that are less than the marker value.
format	String (Optional)	The response format. Valid values are json, xml, or plain. The default is plain. If you append the format=xml or format=json query parameter to the storage account URL, the response shows extended container information serialized in that format. If you append the format=plain query parameter, the response lists the container names separated by newlines.
prefix	String (Optional)	Prefix value. Named items in the response begin with this value.
delimiter	Char (Optional)	Delimiter value, which returns the object names that are nested in the container.

This operation does not accept a request body.

15.3.1.2. Response

Example 15.4. 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": "janeausten"
  },
  {
    "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 <i>(Required)</i>	The name of the container.
count	Int <i>(Required)</i>	The number of objects in the container.
bytes	Int <i>(Required)</i>	The total number of bytes that are stored in Object Storage for the account.

Example 15.5. 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.

15.3.2. 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

15.3.2.1. Request

This table shows the header parameters for the create, update, or delete account metadata request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	Authentication token.
X-Account-Meta-Temp-URL-Key	String <i>(Optional)</i>	The secret key value for temporary URLs.
X-Account-Meta-Temp-URL-Key-2	String <i>(Optional)</i>	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 <i>(Optional)</i>	The account metadata. The <code>{name}</code> 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 <code>X-Account-Meta-{name}</code> header for each metadata item (for each <code>{name}</code>) that you want to add, update, or delete.
Content-Type	String <i>(Optional)</i>	Changes the MIME type for the object.
X-Detect-Content-Type	Boolean <i>(Optional)</i>	If set to <code>true</code> , Object Storage guesses the content type based on the file extension and ignores the value sent in the <code>Content-Type</code> header, if present.

This table shows the URI parameters for the create, update, or delete account metadata request:

Name	Type	Description
<code>{account}</code>	String	The unique name for the account. An account is also known as the project or tenant.

15.3.2.2. Response

This table shows the header parameters for the create, update, or delete account metadata response:

Name	Type	Description
Content-Length	String <i>(Required)</i>	If the operation succeeds, this value is zero (0). If the operation fails, this value is the length of the error text in the response body.
Content-Type	String <i>(Required)</i>	If the operation fails, this value is the MIME type of the error text in the response body.

Name	Type	Description
X-Timestamp	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Trans-Id	Uuid <i>(Required)</i>	A unique transaction identifier for this request. Your service provider might need this value if you report a problem.
Date	DateTime <i>(Required)</i>	<p>The transaction date and time.</p> <p>The date and time stamp format is ISO 8601:</p> <p>CCYY-MM-DDThh:mm:ss±hh:mm</p> <p>The ±hh:mm value, if included, is the time zone as an offset from UTC.</p> <p>For example, 2015-08-27T09:49:58-05:00.</p> <p>A null value indicates that the token never expires.</p>

15.3.3. 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)

15.3.3.1. Request

This table shows the header parameters for the show account metadata request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	Authentication token.
X-Newest	Boolean <i>(Optional)</i>	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

Name	Type	Description
		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.

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.

15.3.3.2. Response

This table shows the header parameters for the show account metadata response:

Name	Type	Description
X-Account-Object-Count	Int <i>(Required)</i>	The number of objects in the account.
X-Account-Container-Count	Int <i>(Required)</i>	The number of containers.
X-Account-Bytes-Used	Int <i>(Required)</i>	The total number of bytes that are stored in Object Storage for the account.
X-Account-Meta-name	String <i>(Optional)</i>	The custom account metadata item, where {name} is the name of the metadata item. One X-Account-Meta-{name} response header appears for each metadata item (for each {name}).
X-Timestamp	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Account-Meta-Temp-URL-Key	String <i>(Optional)</i>	The secret key value for temporary URLs. If not set, this header is not returned in the response.
X-Account-Meta-Temp-URL-Key-2	String <i>(Optional)</i>	A second secret key value for temporary URLs. If not set, this header is not returned in the response.
Content-Length	String <i>(Required)</i>	If the operation succeeds, this value is zero (0). If the operation fails, this value is the length of the error text in the response body.
Content-Type	String <i>(Required)</i>	If the operation fails, this value is the MIME type of the error text in the response body.
X-Trans-Id	Uuid <i>(Required)</i>	A unique transaction identifier for this request. Your service provider might need this value if you report a problem.
Date	DateTime <i>(Required)</i>	The transaction date and time. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. A null value indicates that the token never expires.

15.4. 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.

15.4.1. 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.

Normal response codes: 200, 204

Error response codes: `NotFound` (404)

15.4.1.1. Request

This table shows the header parameters for the show container details and list objects request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	Authentication token.
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-Container-Meta-Temp-URL-Key	String <i>(Optional)</i>	The secret key value for temporary URLs.
X-Container-Meta-Temp-URL-Key-2	String <i>(Optional)</i>	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.

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.

Name	Type	Description
{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 json, xml, or plain. The default is plain. 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.
delimiter	Char (Optional)	Delimiter value, which returns the object names that are nested in the container.
path	String (Optional)	For a string value, returns the object names that are nested in the pseudo path. Equivalent to setting delimiter to / and prefix to the path with a / at the end.

This operation does not accept a request body.

15.4.1.2. Response

Example 15.6. 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: tx26377fe5fab74869825d1-0052d6bdff
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 <i>(Required)</i>	The name of the container.
hash	String <i>(Required)</i>	The MD5 checksum value of the object content.
bytes	Int <i>(Required)</i>	The total number of bytes that are stored in Object Storage for the account.
content_type	String <i>(Required)</i>	The content type of the object.
last_modified	DateTime <i>(Required)</i>	The date and time when the object was last modified. The date and time stamp format is ISO 8601: <code>CCYY-MM-DDThh:mm:ss±hh:mm</code> For example, 2015-08-27T09:49:58-05:00. The ±hh:mm value, if included, is the time zone as an offset from UTC.

Example 15.7. 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>
  </object>
</container>

```

```
<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.

15.4.2. 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: 201, 204

15.4.2.1. Request

This table shows the header parameters for the create container request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	Authentication token.
X-Container-Read	String <i>(Optional)</i>	<p>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.</p> <p>To set the container read ACL:</p> <pre>\$ curl -X {PUT POST} -i -H "X-Auth-Token: TOKEN" -H \ "X-Container-Read: ACL" STORAGE_URL/CONTAINER</pre> <p>For example:</p> <pre>\$ curl -X PUT -i \ -H "X-Auth-Token: 0101010101" \</pre>

Name	Type	Description
		<pre>-H "X-Container-Read: .r:/*" \ http://swift.example.com/v1/AUTH_bob/read_container</pre> <p>In the command, specify the ACL in the X-Container-Read header, as follows:</p> <ul style="list-style-type: none"> • .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-Container-Write	String <i>(Optional)</i>	Sets an ACL that grants write access.
X-Container-Sync-To	String <i>(Optional)</i>	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 <i>(Optional)</i>	Sets the secret key for container synchronization. If you remove the secret key, synchronization is halted.
X-Versions-Location	String <i>(Optional)</i>	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 <i>(Optional)</i>	<p>The container metadata, where {name} is the name of metadata item.</p> <p>You must specify an X-Container-Meta-{name} header for each metadata item (for each {name}) that you want to add or update.</p>
Content-Type	String <i>(Optional)</i>	Changes the MIME type for the object.
X-Detect-Content-Type	Boolean <i>(Optional)</i>	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 <i>(Optional)</i>	The secret key value for temporary URLs.
X-Container-Meta-Temp-URL-Key-2	String <i>(Optional)</i>	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.

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	<p>The unique name for the container.</p> <p>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.</p>

15.4.2.2. Response

This table shows the header parameters for the create container response:

Name	Type	Description
Content-Length	String <i>(Required)</i>	If the operation succeeds, this value is zero (0). If the operation fails, this value is the length of the error text in the response body.
Content-Type	String <i>(Required)</i>	If the operation fails, this value is the MIME type of the error text in the response body.
X-Timestamp	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Trans-Id	Uuid <i>(Required)</i>	A unique transaction identifier for this request. Your service provider might need this value if you report a problem.
Date	DateTime <i>(Required)</i>	The transaction date and time. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. A null value indicates that the token never expires.

15.4.3. 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)

15.4.3.1. Request

This table shows the header parameters for the delete container request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	Authentication token.
X-Container-Meta-Temp-URL-Key	String <i>(Optional)</i>	The secret key value for temporary URLs.
X-Container-Meta-Temp-URL-Key-2	String <i>(Optional)</i>	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.

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.

15.4.3.2. Response

This table shows the header parameters for the delete container response:

Name	Type	Description
Content-Length	String <i>(Required)</i>	If the operation succeeds, this value is zero (0). If the operation fails, this value is the length of the error text in the response body.
Content-Type	String <i>(Required)</i>	If the operation fails, this value is the MIME type of the error text in the response body.
X-Timestamp	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Trans-Id	Uuid <i>(Required)</i>	A unique transaction identifier for this request. Your service provider might need this value if you report a problem.
Date	DateTime <i>(Required)</i>	The transaction date and time. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. A null value indicates that the token never expires.

15.4.4. 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

15.4.4.1. Request

This table shows the header parameters for the create, update, or delete container metadata request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	Authentication token.
X-Container-Read	String <i>(Optional)</i>	<p>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.</p> <p>To set the container read ACL:</p> <pre>\$ curl -X {PUT POST} -i -H "X-Auth-Token: TOKEN" -H "X-Container-Read: ACL" STORAGE_URL/CONTAINER</pre> <p>For example:</p> <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> <p>In the command, specify the ACL in the X-Container-Read header, as follows:</p> <ul style="list-style-type: none"> • .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 <i>(Optional)</i>	Removes the metadata item named {name}. For example, X-Remove-Container-Read removes the X-Container-Read metadata item.
X-Container-Write	String <i>(Optional)</i>	Sets an ACL that grants write access.
X-Container-Sync-To	String <i>(Optional)</i>	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 <i>(Optional)</i>	Sets the secret key for container synchronization. If you remove the secret key, synchronization is halted.
X-Versions-Location	String <i>(Optional)</i>	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 <i>(Optional)</i>	Set to any value to disable versioning.
X-Container-Meta-name	String <i>(Optional)</i>	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-Quota-Bytes	String <i>(Optional)</i>	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 <i>(Optional)</i>	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 <i>(Optional)</i>	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 <i>(Optional)</i>	Changes the MIME type for the object.
X-Detect-Content-Type	Boolean <i>(Optional)</i>	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 <i>(Optional)</i>	The secret key value for temporary URLs.
X-Container-Meta-Temp-URL-Key-2	String <i>(Optional)</i>	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.

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.

15.4.4.2. Response

This table shows the header parameters for the create, update, or delete container metadata response:

Name	Type	Description
Content-Length	String	If the operation succeeds, this value is zero (0). If the operation fails, this value is the length of the error text in the response body.

Name	Type	Description
	(Required)	
Content-Type	String (Required)	If the operation fails, this value is the MIME type of the error text in the response body.
X-Timestamp	Int (Required)	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Trans-Id	Uuid (Required)	A unique transaction identifier for this request. Your service provider might need this value if you report a problem.
Date	DateTime (Required)	The transaction date and time. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. A null value indicates that the token never expires.

15.4.5. 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

15.4.5.1. Request

This table shows the header parameters for the show container metadata request:

Name	Type	Description
X-Auth-Token	String <i>(Optional)</i>	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 <i>(Optional)</i>	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-Container-Meta-Temp-URL-Key	String <i>(Optional)</i>	The secret key value for temporary URLs.
X-Container-Meta-Temp-URL-Key-2	String <i>(Optional)</i>	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.

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

Name	Type	Description
		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.

15.4.5.2. Response

This table shows the header parameters for the show container metadata response:

Name	Type	Description
Content-Length	String <i>(Required)</i>	If the operation succeeds, this value is zero (0). If the operation fails, this value is the length of the error text in the response body.
X-Container-Object-Count	Int <i>(Required)</i>	The number of objects.
Accept-Ranges	String <i>(Required)</i>	The type of ranges that the object accepts.
X-Container-Meta-name	String <i>(Required)</i>	The custom container metadata item, where {name} is the name of the metadata item. One X-Container-Meta-{name} response header appears for each metadata item (for each {name}).
X-Timestamp	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Container-Meta-Quota-Bytes	String <i>(Optional)</i>	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 <i>(Optional)</i>	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-Bytes-Used	Int <i>(Required)</i>	The count of bytes used in total.
X-Container-Read	String <i>(Optional)</i>	The ACL that grants read access. If not set, this header is not returned by this operation.
X-Container-Write	String <i>(Optional)</i>	The ACL that grants write access. If not set, this header is not returned by this operation.
X-Container-Sync-To	String <i>(Optional)</i>	The destination for container synchronization. If not set, this header is not returned by this operation.
X-Container-Sync-Key	String <i>(Optional)</i>	The secret key for container synchronization. If not set, this header is not returned by this operation.
X-Versions-Location	String <i>(Required)</i>	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.
Content-Type	String <i>(Required)</i>	If the operation fails, this value is the MIME type of the error text in the response body.
X-Trans-Id	Uuid <i>(Required)</i>	A unique transaction identifier for this request. Your service provider might need this value if you report a problem.

Name	Type	Description
Date	DateTime <i>(Required)</i>	<p>The transaction date and time.</p> <p>The date and time stamp format is ISO 8601:</p> <pre>CCYY-MM-DDThh:mm:ss±hh:mm</pre> <p>The <code>±hh:mm</code> value, if included, is the time zone as an offset from UTC.</p> <p>For example, <code>2015-08-27T09:49:58-05:00</code>.</p> <p>A <code>null</code> value indicates that the token never expires.</p>
X-Container-Meta-Temp-URL-Key	String <i>(Optional)</i>	The secret key value for temporary URLs. If not set, this header is not returned in the response.
X-Container-Meta-Temp-URL-Key-2	String <i>(Optional)</i>	A second secret key value for temporary URLs. If not set, this header is not returned in the response.

15.5. 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.

15.5.1. 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>
```

Normal response codes: 200

Error response codes: NotFound (404)

15.5.1.1. Request

This table shows the header parameters for the get object content and metadata request:

Name	Type	Description
X-Auth-Token	String <i>(Optional)</i>	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 <i>(Optional)</i>	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.
Range	String <i>(Optional)</i>	<p>The ranges of content to get.</p> <p>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.</p> <p>The types of range specifications are:</p> <ul style="list-style-type: none"> • Byte range specification. Use FIRST_BYTE_OFFSET to specify the start of the data range, and LAST_BYTE_OFFSET to specify the end. You can omit the LAST_BYTE_OFFSET and if you do, the value defaults to the offset of the last byte of data. • Suffix byte range specification. Use LENGTH bytes to specify the length of the data range. <p>The following forms of the header specify the following ranges of data:</p> <ul style="list-style-type: none"> • Range: bytes=-5. The last five bytes. • Range: bytes=10-15. The five bytes of data after a 10-byte offset. • Range: bytes=10-15,-5. A multi-part response that contains the last five bytes and the five bytes of data after a 10-byte offset. The Content-Type of the response is then multipart/byteranges. • Range: bytes=4-6. Bytes 4 to 6 inclusive. • Range: bytes=2-2. Byte 2, the third byte of the data. • Range: bytes=6-. Byte 6 and after. • Range: bytes=1-3,2-5. A multi-part response that contains bytes 1 to 3 inclusive, and bytes 2 to 5 inclusive. The Content-Type of the response is then multipart/byteranges.
If-Match	String <i>(Optional)</i>	See http://www.ietf.org/rfc/rfc2616.txt .
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.
If-Modified-Since	Date <i>(Optional)</i>	See Request for Comments: 2616 .
If-Unmodified-Since	Date <i>(Optional)</i>	See Request for Comments: 2616 .

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.

Name	Type	Description
		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>	Used with temporary URLs to override 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 be saved. 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 header for dynamic large objects or in the response body for static large objects.

15.5.1.2. Response

This table shows the header parameters for the get object content and metadata response:

Name	Type	Description
Content-Length	String <i>(Required)</i>	The length of the object content in the response body, in bytes.
Accept-Ranges	String <i>(Required)</i>	The type of ranges that the object accepts.
Last-Modified	DateTime <i>(Required)</i>	The date and time when the object was created or its metadata was changed. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00.
X-Timestamp	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.

Name	Type	Description
ETag	String <i>(Required)</i>	<p>For objects smaller than 5 GB, this value is the MD5 checksum of the object content. The value is not quoted.</p> <p>For manifest objects, this value is the MD5 checksum of the concatenated string of MD5 checksums and ETags for each of the segments in the manifest, and not the MD5 checksum of the content that was downloaded. Also the value is enclosed in double-quote characters.</p> <p>You are strongly recommended to compute the MD5 checksum of the response body as it is received and compare this value with the one in the ETag header. If they differ, the content was corrupted, so retry the operation.</p>
Content-Type	String <i>(Required)</i>	The MIME type of the object.
Content-Encoding	String <i>(Optional)</i>	<p>If set, the value of the Content-Encoding metadata.</p> <p>If not set, this header is not returned by this operation.</p>
Content-Disposition	String <i>(Optional)</i>	<p>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.</p> <p>If not set, this header is not returned by this operation.</p>
X-Delete-At	String <i>(Optional)</i>	<p>If set, the date and time in UNIX Epoch time stamp format when the system deletes the object.</p> <p>For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.</p> <p>If not set, this header is not returned by this operation.</p>
X-Object-Meta-name	String <i>(Required)</i>	<p>The custom object metadata item, where {name} is the name of the metadata item.</p> <p>One X-Object-Meta-{name} response header appears for each metadata item (for each {name}).</p>
X-Object-Manifest	String <i>(Optional)</i>	If set, to this is a dynamic large object manifest object. The value is the container and object name prefix of the segment objects in the form container/prefix.
X-Static-Large-Object	Boolean <i>(Required)</i>	Set to True if this object is a static large object manifest object.
X-Trans-Id	Uuid <i>(Required)</i>	A unique transaction identifier for this request. Your service provider might need this value if you report a problem.
Date	DateTime <i>(Required)</i>	<p>The transaction date and time.</p> <p>The date and time stamp format is ISO 8601:</p> <p style="background-color: #f0f0f0; padding: 2px;">CCYY-MM-DDThh:mm:ss±hh:mm</p> <p>The ±hh:mm value, if included, is the time zone as an offset from UTC.</p> <p>For example, 2015-08-27T09:49:58-05:00.</p> <p>A null value indicates that the token never expires.</p>

15.5.2. 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/janeausten/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/janeausten/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)

15.5.2.1. Request

This table shows the header parameters for the create or replace object request:

Name	Type	Description
X-Object-Manifest	String <i>(Optional)</i>	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 <i>(Optional)</i>	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 <i>(Optional)</i>	Set to the length of the object content. Do not set if chunked transfer encoding is being used.
Transfer-Encoding	String <i>(Optional)</i>	Set to <code>chunked</code> to enable chunked transfer encoding. If used, do not set the Content-Length header to a non-zero value.
Content-Type	String <i>(Optional)</i>	Changes the MIME type for the object.
X-Detect-Content-Type	Boolean <i>(Optional)</i>	If set to <code>true</code> , 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 <i>(Optional)</i>	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.
ETag	String <i>(Optional)</i>	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 <i>(Optional)</i>	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 <i>(Optional)</i>	If set, the value of the Content-Encoding metadata.
X-Delete-At	Int <i>(Optional)</i>	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 <i>(Optional)</i>	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 <i>(Optional)</i>	The object metadata, where <code>{name}</code> is the name of the metadata item.

Name	Type	Description
		You must specify an <code>x-Object-Meta-{name}</code> header for each meta-data item (for each <code>{name}</code>) that you want to add or update.
If-None-Match	String <i>(Optional)</i>	In combination with <code>Expect: 100-Continue</code> , specify an <code>If-None-Match: *</code> header to query whether the server already has a copy of the object before any data is sent.

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 <code>?multipart-manifest=put</code> , 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 <i>(Optional)</i>	Used with temporary URLs to override the default file name. Object Storage generates a default file name for <code>GET</code> temporary URLs that is based on the object name. Object Storage returns this value in the <code>Content-Disposition</code> response header. Browsers can interpret this file name value as a file attachment to be saved. For more information about temporary URLs, see Temporary URL middleware .

15.5.2.2. Response

This table shows the header parameters for the create or replace object response:

Name	Type	Description
Content-Length	String <i>(Required)</i>	If the operation succeeds, this value is zero (0). If the operation fails, this value is the length of the error text in the response body.
ETag	String <i>(Required)</i>	For objects smaller than 5 GB, this value is the MD5 checksum of the uploaded object content. The value is not quoted. If you supplied an <code>ETag</code> request header and the operation was successful, the values are the same.

Name	Type	Description
		If you did not supply an ETag request header, check the ETag response header value against the object content you have just uploaded. For static large objects, this value is the MD5 checksum of the concatenated string of MD5 checksums and ETags for each of the segments in the manifest, and not the MD5 checksum of the content that was uploaded. Also the value is enclosed in double-quotes. For dynamic large objects, the value is the MD5 checksum of the empty string.
Content-Type	String <i>(Required)</i>	The MIME type of the object.
X-Timestamp	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Trans-Id	Uuid <i>(Required)</i>	A unique transaction identifier for this request. Your service provider might need this value if you report a problem.
Date	DateTime <i>(Required)</i>	The transaction date and time. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. A null value indicates that the token never expires.

15.5.3. 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: txdcb481ad49d24e9a81107-0052d97501
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: txdcb481ad49d24e9a81107-0052d97501
Date: Fri, 17 Jan 2014 18:22:57 GMT
```



Note

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

15.5.3.1. Request

This table shows the header parameters for the copy object request:

Name	Type	Description
x-Auth-Token	String <i>(Optional)</i>	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 <i>(Required)</i>	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 <i>(Optional)</i>	Changes the MIME type for the object.
Content-Encoding	String <i>(Optional)</i>	If set, the value of the Content-Encoding metadata.
Content-Disposition	String <i>(Optional)</i>	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 <i>(Optional)</i>	The object metadata, where {name} is the name of the metadata item. You must specify an x-Object-Meta-{name} header for each metadata item (for each {name}) that you want to add or update.
x-Fresh-Metadata	Boolean <i>(Optional)</i>	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.

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

Name	Type	Description
		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.

15.5.3.2. Response

This table shows the header parameters for the copy object response:

Name	Type	Description
Content-Length	String <i>(Required)</i>	If the operation succeeds, this value is zero (0). If the operation fails, this value is the length of the error text in the response body.
X-Copied-From-Last-Modified	Int <i>(Optional)</i>	For a copied object, the date and time in UNIX Epoch time stamp format when the container and object name from which the new object was copied was last modified. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Copied-From	String <i>(Optional)</i>	For a copied object, shows the container and object name from which the new object was copied. The value is in the {container}/{object} format.
Last-Modified	DateTime <i>(Required)</i>	The date and time when the object was created or its metadata was changed. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00.
ETag	String <i>(Required)</i>	The MD5 checksum of the copied object content. The value is not quoted.
Content-Type	String <i>(Required)</i>	The MIME type of the object.
X-Timestamp	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Object-Meta-name	String <i>(Required)</i>	The custom object metadata item, where {name} is the name of the metadata item. One X-Object-Meta-{name} response header appears for each metadata item (for each {name}).
X-Trans-Id	Uuid <i>(Required)</i>	A unique transaction identifier for this request. Your service provider might need this value if you report a problem.
Date	DateTime <i>(Required)</i>	The transaction date and time. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. A null value indicates that the token never expires.

15.5.4. 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: tx36c7606fcfd1843f59167c-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.

Error response codes: 400, 500, ...

15.5.4.1. Request

This table shows the header parameters for the delete object request:

Name	Type	Description
X-Auth-Token	String <i>(Optional)</i>	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).

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 before the first period (.) character.

Name	Type	Description
		cause 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.

15.5.4.2. Response

This table shows the header parameters for the delete object response:

Name	Type	Description
Content-Length	String <i>(Required)</i>	If the operation succeeds, this value is zero (0). If the operation fails, this value is the length of the error text in the response body.
Content-Type	String <i>(Required)</i>	The MIME type of the object.
X-Timestamp	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Trans-Id	Uuid <i>(Required)</i>	A unique transaction identifier for this request. Your service provider might need this value if you report a problem.
Date	DateTime <i>(Required)</i>	The transaction date and time. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. A null value indicates that the token never expires.

15.5.5. 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: tx37ea34dc1ed48ca9bc7d-0052d84b6f
Date: Thu, 16 Jan 2014 21:13:19 GMT
```

If the request succeeds, the operation returns the 204 response code.

Normal response codes: 204

15.5.5.1. Request

This table shows the header parameters for the show object metadata request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	Authentication token.
X-Newest	Boolean <i>(Optional)</i>	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.

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

Name	Type	Description
		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 <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>	Used with temporary URLs to override 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 be saved. For more information about temporary URLs, see Temporary URL middleware .

15.5.5.2. Response

This table shows the header parameters for the show object metadata response:

Name	Type	Description
Last-Modified	DateTime <i>(Required)</i>	The date and time when the object was created or its metadata was changed. The date and time stamp format is ISO 8601 : CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00.
Content-Length	String <i>(Required)</i>	The length of the object content in the response body, in bytes.
Content-Length	String <i>(Required)</i>	HEAD operations do not return content. However, in this operation the value in the Content-Length header is not the size of the response body. Instead it contains the size of the object, in bytes.
Content-Type	String <i>(Required)</i>	The MIME type of the object.
X-Timestamp	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
ETag	String <i>(Required)</i>	For objects smaller than 5 GB, this value is the MD5 checksum of the object content. The value is not quoted.

Name	Type	Description
		<p>For manifest objects, this value is the MD5 checksum of the concatenated string of MD5 checksums and ETags for each of the segments in the manifest, and not the MD5 checksum of the content that was downloaded. Also the value is enclosed in double-quote characters.</p> <p>You are strongly recommended to compute the MD5 checksum of the response body as it is received and compare this value with the one in the ETag header. If they differ, the content was corrupted, so retry the operation.</p>
Content-Encoding	String <i>(Optional)</i>	<p>If set, the value of the Content-Encoding metadata.</p> <p>If not set, this header is not returned by this operation.</p>
Content-Disposition	String <i>(Optional)</i>	<p>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.</p> <p>If not set, this header is not returned by this operation.</p>
X-Delete-At	String <i>(Optional)</i>	<p>If set, the date and time in UNIX Epoch time stamp format when the system deletes the object.</p> <p>For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.</p> <p>If not set, this header is not returned by this operation.</p>
X-Object-Manifest	String <i>(Optional)</i>	<p>If set, to 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>.</p>
X-Object-Meta-name	String <i>(Required)</i>	<p>The custom object metadata item, where <code>{name}</code> is the name of the metadata item.</p> <p>One X-Object-Meta-{name} response header appears for each metadata item (for each <code>{name}</code>).</p>
X-Static-Large-Object	Boolean <i>(Required)</i>	<p>Set to True if this object is a static large object manifest object.</p>
X-Trans-Id	Uuid <i>(Required)</i>	<p>A unique transaction identifier for this request. Your service provider might need this value if you report a problem.</p>
Date	DateTime <i>(Required)</i>	<p>The transaction date and time.</p> <p>The date and time stamp format is ISO 8601:</p> <p style="background-color: #f0f0f0; padding: 2px;"><code>CCYY-MM-DDThh:mm:ss±hh:mm</code></p> <p>The <code>±hh:mm</code> value, if included, is the time zone as an offset from UTC.</p> <p>For example, <code>2015-08-27T09:49:58-05:00</code>.</p> <p>A null value indicates that the token never expires.</p>

15.5.6. 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

15.5.6.1. Request

This table shows the header parameters for the create or update object metadata request:

Name	Type	Description
X-Auth-Token	String <i>(Optional)</i>	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 <i>(Optional)</i>	The object metadata, where {name} is the name of the metadata item. You must specify an X-Object-Meta-{name} header for each metadata item (for each {name}) that you want to add or update.
X-Delete-At	Int <i>(Optional)</i>	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 <i>(Optional)</i>	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 <i>(Optional)</i>	If set, the value of the Content-Encoding metadata.
X-Delete-After	Int <i>(Optional)</i>	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 <i>(Optional)</i>	Changes the MIME type for the object.
X-Detect-Content-Type	Boolean <i>(Optional)</i>	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.

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.

15.5.6.2. Response

This table shows the header parameters for the create or update object metadata response:

Name	Type	Description
Content-Length	String <i>(Required)</i>	If the operation succeeds, this value is zero (0). If the operation fails, this value is the length of the error text in the response body.
Content-Type	String <i>(Required)</i>	The MIME type of the object.
X-Timestamp	Int <i>(Required)</i>	The date and time in UNIX Epoch time stamp format when the account, container, or object was initially created as a current version. For example, 1440619048 is equivalent to Mon, Wed, 26 Aug 2015 19:57:28 GMT.
X-Trans-Id	Uuid <i>(Required)</i>	A unique transaction identifier for this request. Your service provider might need this value if you report a problem.
Date	DateTime <i>(Required)</i>	The transaction date and time. The date and time stamp format is ISO 8601 : <code>CCYY-MM-DDThh:mm:ss±hh:mm</code> The <code>±hh:mm</code> value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. A null value indicates that the token never expires.

16. 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 describing 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{?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.

16.1. 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.

16.2. API versions

Method	URI	Description
GET	/	Lists all Orchestration API versions.

16.2.1. List versions

Method	URI	Description
GET	/	Lists all Orchestration API versions.

Normal response codes: 200

16.2.1.1. Request

This operation does not accept a request body.

16.2.1.2. Response

Example 16.1. List versions: JSON response

```
{
  "versions": [
    {
      "status": "CURRENT",
      "id": "v1.0",
      "links": [
        {
          "href": "http://23.253.228.211:8000/v1/",
          "rel": "self"
        }
      ]
    }
  ]
}
```

16.3. 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 describing 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.

16.3.1. 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	

16.3.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.

Example 16.2. 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 -x\nnecho \"hello world\" > /root/hello-world.txt\n"
                }
            }
        }
    }
}
```

```
    } ,  
    "timeout_mins": 60  
}
```

16.3.1.2. Response

Example 16.3. Create stack: JSON response

```
{  
    "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"  
            }  
        ]  
    }  
}
```

16.3.2. 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	

16.3.2.1. Request

This table shows the URI parameters for the adopt stack request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.

Example 16.4. Adopt stack: JSON request

```
{
  "adopt_stack_data": {
    "action": "CREATE",
    "id": "bxxxxx4-0xx2-4xx1-axx6-xxxxxxxxc",
    "name": "teststack",
    "resources": {
      "MyServer": {
        "action": "CREATE",
        "metadata": {},
        "name": "MyServer",
        "resource_data": {},
        "resource_id": "cxxxx3-dxx3-4xx-bxx2-3xxxxxxxxa",
        "status": "COMPLETE",
        "type": "OS::Trove::Instance"
      }
    },
    "status": "COMPLETE",
    "template": {}
  },
  "stack_name": "{stack_name}",
  "template_url": "{template_url}",
  "timeout_mins": "{timeout_mins}"
}
```

16.3.2.2. Response

Example 16.5. Adopt stack: JSON response

```
{
```

```
"action": "CREATE",
"id": "46c927bb-671a-43db-a29c-16a2610865ca",
"name": "trove",
"resources": {
    "mysql_server": {
        "action": "CREATE",
        "metadata": {},
        "name": "mysql_server",
        "resource_data": {},
        "resource_id": "74c5be7e-3e62-41e7-b455-93d1c32d56e3",
        "status": "COMPLETE",
        "type": "OS::Trove::Instance"
    }
},
"status": "COMPLETE",
"template": {
    "heat-template-version": "2013-05-23",
    "description": "MySQL server instance",
    "parameters": {
        "instance_name": {
            "description": "The database instance name",
            "type": "string"
        }
    },
    "resources": {
        "mysql_server": {
            "properties": {
                "databases": [
                    {
                        "name": "testdbonetwo"
                    }
                ],
                "flavor": "m1.medium",
                "name": "test-trove-db",
                "size": 30,
                "users": [
                    {
                        "databases": [
                            "testdbonetwo"
                        ],
                        "name": "testuser",
                        "password": "testpass123"
                    }
                ]
            },
            "type": "OS::Trove::Instance"
        }
    }
}
}
```

16.3.3. 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	

16.3.3.1. Request

This table shows the URI parameters for the list stack data request:

Name	Type	Description
{tenant_id}	String	The ID 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	String <i>(Optional)</i>	Filters the stack list by a stack ID. Use this filter multiple times to filter by multiple IDs.
status	String <i>(Optional)</i>	Filters the stack list by a status. Use this filter multiple times to filter by multiple statuses.
name	String <i>(Optional)</i>	Filters the stack list by a name. Use this filter multiple times to filter by multiple names.
action	String <i>(Optional)</i>	Filters the stack list by an action. Use this filter multiple times to filter by multiple actions.
tenant	String <i>(Optional)</i>	Filters the stack list by a tenant. Use this filter multiple times to filter by multiple tenants.
username	String <i>(Optional)</i>	Filters the stack list by a user name. Use this filter multiple times to filter by multiple user names.
owner_id	String <i>(Optional)</i>	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 <i>(Optional)</i>	Requests a page size of returned items from the query. 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

Name	Type	Description
		response as the <code>marker</code> parameter value in a subsequent limited request.
marker	String <i>(Optional)</i>	Specifies 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.
show_deleted	String <i>(Optional)</i>	Specifies whether to include deleted stacks in the list. Default is <code>False</code> , which excludes deleted stacks from the list.
show_nested	String <i>(Optional)</i>	Specifies whether to include nested stacks in the list. Default is <code>False</code> , which excludes nested stacks from the list.
sort_keys	String <i>(Optional)</i>	Sorts the stack list by <code>name</code> , <code>status</code> , <code>created_at</code> , or <code>updated_at</code> key.
tags	String <i>(Optional)</i>	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 <i>(Optional)</i>	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 <i>(Optional)</i>	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 <i>(Optional)</i>	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 <i>(Optional)</i>	The sort direction of the list. A valid value is <code>asc</code> (ascending) or <code>desc</code> (descending).
global_tenant	String <i>(Optional)</i>	Specifies whether to include stacks from all tenants in the stack list. Specify policy requirements in the Orchestration <code>policy.json</code> file. Default is <code>False</code> .
with_count	String <i>(Optional)</i>	Specifies whether to include a <code>count</code> key in the response. The <code>count</code> 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.

16.3.3.2. Response

Example 16.6. 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": ""
    }
]
}
```

16.3.4. 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	

16.3.4.1. Request

This table shows the URI parameters for the preview stack request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.

Example 16.7. 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\" > /root/hello-world.txt\n"
                }
            }
        }
    },
    "timeout_mins": 60
}
```

{

16.3.4.2. Response

Example 16.8. 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/  
6e18cc2bdbbeb48a5basad2dc499f6804/stacks/test_stack/None",  
                "rel": "self"  
            }  
        ],  
        "notification_topics": [],  
        "parameters": {  
            "OS::project_id": "6e18cc2bdbbeb48a5basad2dc499f6804",  
            "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"  
                        }  
                    ]  
                }  
            }  
        ]  
    }  
}
```

```
        "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": "6e18cc2bdbbeb48a5b3cad2dc499f6804"
},
"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": "6e18cc2bdbbeb48a5b3cad2dc499f6804"
},
"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",
        "image": null,
        "key_name": null,
        "metadata": {},
        "networks": [
            {
                "id": "net-1",
                "label": "private"
            }
        ],
        "osapi_image": null,
        "osapi_key": null,
        "osapi_metadata": {},
        "osapi_network": null,
        "osapi_tenant": null,
        "osapi_user": null,
        "ramdisk": null
    }
}
```

```
        "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": "6e18cc2bdbbeb48a3433cad2dc499sdf32234"
    },
    "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": "6e18cc2bdbbeb48a3433cad2dc499sdf32234"
    },
    "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
}
```

16.3.5. 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

16.3.5.1. Request

This table shows the URI parameters for the preview stack update request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

Example 16.9. 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"
  }
}
```

16.3.5.2. Response

Example 16.10. Preview stack update: JSON response

```
{
  "unchanged": [
    {
      "updated_time": "datetime",
      "status": "OK"
    }
  ]
}
```

```
        "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}"
}
```

```
        }  
    ]  
}
```

16.3.6. 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 you can perform **PUT** and **DELETE** operations on a current stack. Set your client to follow redirects. Note that 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 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	

16.3.6.1. Request

This table shows the URI parameters for the find stack request:

Name	Type	Description
{tenant_id}	String	The ID 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.

16.3.7. 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

16.3.7.1. Request

This table shows the URI parameters for the find stack resources request:

Name	Type	Description
{tenant_id}	String	The ID 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.

16.3.8. 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	

16.3.8.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

This operation does not accept a request body.

16.3.8.2. Response

Example 16.11. 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-afal-e254afc4a435",
      "OS::stack_id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
      "OS::stack_name": "simple_stack"
    },
    "status": "CREATE_IN_PROGRESS"
  }
}
```

```
        "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": ""
    }
}
```

16.3.9. 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	

16.3.9.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

Example 16.12. 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\nnecho \"hello world\" > /root/hello-world.txt\n"
                }
            }
        }
    },
    "parameters": {
        "flavor": "m1.small"
    }
}
```

```
    }  
}
```

16.3.10. 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	

16.3.10.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

This operation does not accept a request body.

16.3.11. 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 describing the stack and its resources.

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	

16.3.11.1. Request

This table shows the URI parameters for the abandon stack request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

This operation does not accept a request body.

16.3.11.2. Response

Example 16.13. 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": {}
        }
    }
}
```

16.3.12. 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

16.3.12.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

Example 16.14. Snapshot stack: JSON request

```
{
    "name": "vol_snapshot"
}
```

16.3.12.2. Response

Example 16.15. 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"
}
```

16.3.13. List snapshots

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots	Lists snapshots for a stack.

Normal response codes: 200

16.3.13.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

This operation does not accept a request body.

16.3.13.2. Response

Example 16.16. List snapshots: JSON response

```
{
  "snapshots": [
    {
      "id": "7c4e1ef4-bf1b-41ab-a0c8-ce01f4ffdfal",
      "name": "vol_snapshot",
      "status": "IN_PROGRESS",
      "status_reason": null,
      "creation_time": "2015-08-04T20:57:55Z",
      "data": null
    }
  ]
}
```

16.3.14. 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

16.3.14.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.
{snapshot_id}	String	The snapshot ID.

This operation does not accept a request body.

16.3.14.2. Response

Example 16.17. Show snapshot: JSON response

```
{
  "snapshot": {
    "id": "7c4e1ef4-bf1b-41ab-a0c8-ce01f4ffdf1",
    "name": "vol_snapshot",
    "status": "COMPLETE",
    "status_reason": "Stack SNAPSHOT completed successfully",
    "creation_time": "2015-08-04T20:57:55Z",
    "data": {
      "status": "COMPLETE",
      "name": "stack_voll",
      "stack_user_project_id": "ffffa11067b1c48129ddfb78fba2bf09f",
      "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": {}
    }
}
}
```

16.3.15. 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

16.3.15.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.
{snapshot_id}	String	The snapshot ID.

This operation does not accept a request body.

16.3.16. 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

16.3.16.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.
{snapshot_id}	String	The snapshot ID.

This operation does not accept a request body.

16.3.17. List outputs

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs	Lists outputs for a stack.

Normal response codes: 200

16.3.17.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

This operation does not accept a request body.

16.3.17.2. Response

Example 16.18. List outputs: JSON response

```
{
    "outputs": [
        {
            "output_key": "output name",
            "output_value": "output value",
            "description": "output description",
            "output_error": null
        }
    ]
}
```

16.3.18. 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

16.3.18.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.
{output_key}	String	The key of a stack output.

This operation does not accept a request body.

16.3.18.2. Response

Example 16.19. Show output: JSON response

```
{
  "output": {
    "output_key": "output_name",
    "output_value": "output_value",
    "description": "output description",
    "output_error": null
  }
}
```

16.4. 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	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.

16.4.1. Suspend stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Suspends a stack.

Normal response codes: 201

16.4.1.1. Request

This table shows the URI parameters for the suspend stack request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

Example 16.20. Suspend stack: JSON request

```
{
    "suspend": null
}
```

16.4.2. Resume stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Resumes a suspended stack.

Normal response codes: 201

16.4.2.1. Request

This table shows the URI parameters for the resume stack request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

Example 16.21. Resume stack: JSON request

```
{
    "resume": null
}
```

16.4.3. 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: 201

16.4.3.1. Request

This table shows the URI parameters for the cancel stack update request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

Example 16.22. Cancel stack update: JSON request

```
{
    "cancel_update": null
}
```

16.4.4. 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: 201

16.4.4.1. Request

This table shows the URI parameters for the check stack resources request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

Example 16.23. Check stack resources: JSON request

```
{
    "check": null
}
```

16.5. 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.

16.5.1. 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	

16.5.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

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

Name	Type	Description
nested_depth	String <i>(Optional)</i>	Includes resources from nested stacks up to the nested_depth levels of recursion.
with_detail	String <i>(Optional)</i>	Enables detailed resource information for each resource in list of resources.

This operation does not accept a request body.

16.5.1.2. Response

Example 16.24. 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"
    }
]
```

16.5.2. 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	

16.5.2.1. Request

This table shows the URI parameters for the show resource data request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.
{resource_name}	String	The name of a resource in the stack.

This operation does not accept a request body.

16.5.2.2. Response

Example 16.25. 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"
}
}
```

16.5.3. 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

16.5.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.
{resource_name}	String	The name of a resource in the stack.

This operation does not accept a request body.

16.5.3.2. Response

Example 16.26. Show resource metadata: JSON response

```
{
  "metadata": {
    "some_key": "some_value",
    "some_other_key": "some_other_value"
  }
}
```

16.5.4. 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

16.5.4.1. Request

This table shows the URI parameters for the send a signal to a resource request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.
{resource_name}	String	The name of a resource in the stack.

This operation does not accept a request body.

16.6. 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.

16.6.1. 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

16.6.1.1. Request

This table shows the URI parameters for the find stack events request:

Name	Type	Description
{tenant_id}	String	The ID 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.

16.6.2. 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	

16.6.2.1. Request

This table shows the URI parameters for the list stack events request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

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 returned items from the query. 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)	Specifies 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

Name	Type	Description
		from the response as the <code>marker</code> parameter value in a subsequent limited request.
sort_keys	String <i>(Optional)</i>	Sorts the list by the <code>resource_type</code> or <code>created_at</code> key.
sort_dir	String <i>(Optional)</i>	The sort direction of the list. A valid value is <code>asc</code> (ascending) or <code>desc</code> (descending).

This operation does not accept a request body.

16.6.2.2. Response

Example 16.27. 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/474bfdf0-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": "474bfdf0-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-blaf-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"
}
]
```

16.6.3. 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	

16.6.3.1. Request

This table shows the URI parameters for the list resource events request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.
{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 <i>(Optional)</i>	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 <i>(Optional)</i>	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 <i>(Optional)</i>	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 <i>(Optional)</i>	Requests a page size of returned items from the query. 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>	Specifies 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	Type	Description
sort_keys	String <i>(Optional)</i>	Sorts the list by the resource_type or created_at key.
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.

16.6.3.2. Response

Example 16.28. 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/474bfdf0-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": "474bfdf0-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"  
}  
]  
}
```

16.6.4. 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.

```
{
  "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"
  }
}
```

Normal response codes: 200

16.6.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.
{resource_name}	String	The name of a resource in the stack.

Name	Type	Description
{event_id}	String	The ID of an event that is related to the resource in the stack.

This operation does not accept a request body.

16.6.4.2. Response

This operation does not return a response body.

16.7. 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{?support_status}	Lists all supported template resource types.

16.7.1. 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	

16.7.1.1. Request

This table shows the URI parameters for the get stack template request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{stack_name}	String	The name of a stack.
{stack_id}	String	The stack ID.

This operation does not accept a request body.

16.7.1.2. Response

Example 16.29. 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"
        }
    }
}
```

16.7.2. 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
400	Bad Request	
401	Unauthorized	
404	Not Found	
500	Internal Server Error	

16.7.2.1. Request

This table shows the URI parameters for the list template versions request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

16.7.3. Validate template

Method	URI	Description
POST	/v1/{tenant_id}/validate	Validates a template.

Normal response codes: 200

16.7.3.1. Request

This table shows the URI parameters for the validate template request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.

Example 16.30. Validate template: JSON request

```
{
    "template_url": "/PATH_TO_HEAT_TEMPLATES/WordPress_Single_Instance.
template"
}
```

16.7.3.2. Response

Example 16.31. 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"
        ]
    }
]
```

16.7.4. 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	

16.7.4.1. Request

This table shows the URI parameters for the show resource template request:

Name	Type	Description
{tenant_id}	String	The ID 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.

16.7.4.2. Response

Example 16.32. Show resource template: JSON response

```
{
    "HeatTemplateFormatVersion": "2012-12-12",
    "Outputs": {
        "private_key": {
            "Description": "The private key if it has been saved.",
            "Value": "{\"Fn::GetAtt\": [\"KeyValuePair\", \"private_key\"]}"
        },
        "public_key": {
            "Description": "The public key.",
            "Value": "{\"Fn::GetAtt\": [\"KeyValuePair\", \"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::KeyValuePair"
    }
}
}
```

16.7.5. 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	

16.7.5.1. Request

This table shows the URI parameters for the show resource schema request:

Name	Type	Description
{tenant_id}	String	The ID 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.

16.7.5.2. Response

Example 16.33. 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"
    }
}
```

16.7.6. List resource types

Method	URI	Description
GET	/v1/{tenant_id}/resource_types{?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	

16.7.6.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID 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
support_status	String <i>(Optional)</i>	Filters the 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.

16.7.6.2. Response

Example 16.34. 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",
        "AWS::CloudFormation::WaitCondition"
    ]
}
```

```
"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"
]
}
```

16.8. Build info

Method	URI	Description
GET	/v1/{tenant_id}/build_info	Shows build information for an Orchestration deployment.

16.8.1. Show build information

Method	URI	Description
GET	/v1/{tenant_id}/build_info	Shows build information for an Orchestration deployment.

Normal response codes: 200

16.8.1.1. Request

This table shows the URI parameters for the show build information request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

16.8.1.2. Response

Example 16.35. Show build information: JSON response

```
{
    "api": {
        "revision": "{api_build_revision}"
    },
    "engine": {
        "revision": "{engine_build_revision}"
    }
}
```

16.9. 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.
DELETE	/v1/{tenant_id}/software_deployments/{deployment_id}	Deletes a software deployment.

16.9.1. 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	

16.9.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.

Example 16.36. 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\nnecho \"Writing to /tmp/$bar\"\nnecho $foo > /tmp/$bar\nnecho -n \"The file /tmp/$bar contains `cat /tmp/$bar` for server $deploy_server_id during $deploy_action\" > $heat_outputs_path.result\nnecho \"Written to /tmp/$bar\"\nnecho \"Output to stderr\" 1>&2",
  "options": null
}
```

16.9.1.2. Response

Example 16.37. 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"  
    }  
}
```

16.9.2. 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	

16.9.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{config_id}	String	The configuration ID.

This operation does not accept a request body.

16.9.2.2. Response

Example 16.38. 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\nnecho \"Writing to /tmp/$bar\"\nnecho $foo > /tmp/$bar\nnecho -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\"\nnecho \"Output to stderr\" 1>&2",
    "options": null
}
```

16.9.3. Delete config

Method	URI	Description
DELETE	/v1/{tenant_id}/software_configs/{config_id}	Deletes a software configuration.

Normal response codes: 204

16.9.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{config_id}	String	The configuration ID.

This operation does not accept a request body.

16.9.4. 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	

16.9.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

16.9.4.2. Response

Example 16.39. 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+ cat /tmp/barmy\n+ echo -n The file /tmp/barmy contains fu for server ec14c864-096e-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 ec14c864-096e-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"
        }
    ]
}
```

```
        ]  
    }
```

16.9.5. Create deployment

Method	URI	Description
POST	/v1/{tenant_id}/software_deployments	Creates a software deployment.

Normal response codes: 200

16.9.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.

Example 16.40. 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"
}
```

16.9.5.2. Response

Example 16.41. 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"
    }
}
```

16.9.6. 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

16.9.6.1. Request

This table shows the URI parameters for the show server configuration metadata request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{server_id}	String	The ID of the server for which to fetch configuration metadata.

This operation does not accept a request body.

16.9.6.2. Response

Example 16.42. 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
%2BNbg86mqwSOUf1GLuKPO7KaD82PiGpL4ig9Q114%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\n# echo \"Writing to /tmp/$bar\"\n# echo $foo
> /tmp/$bar\n# echo -n \"The file /tmp/$bar contains `cat /tmp/$bar` for server
$deploy_server_id during $deploy_action\" > $heat_outputs_path.result\n# echo
\"Written to /tmp/$bar\"\n# echo \"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-adf9elac70fc",
            "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-adf9elac70fc%2Fresources%2Fother_deployment?
Timestamp=2014-03-19T20%3A30%3A59Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=
7b761482f8254946bcd3d5ccb36fe939&SignatureVersion=2&Signature=giMfv%2BhrAw6y
%2FCMKQIQz2Iho5PkAj5%2BfP5YsL6rul3o%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": "2015-01-31T16:14:13Z",
    "updated_time": "2015-01-31T16:18:19Z",
    "config": "#!/bin/sh -x\nnecho \"Writing to /tmp/$bar\"\nnecho $foo
> /tmp/$bar\nnecho -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"
    }
]
```

}

16.9.7. Show deployment details

Method	URI	Description
GET	/v1/{tenant_id}/software_deployments/{deployment_id}	Shows details for a software deployment.

Normal response codes: 200

16.9.7.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{deployment_id}	String	The deployment ID.

This operation does not accept a request body.

16.9.7.2. Response

Example 16.43. 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"
    }
}
```

16.9.8. Update deployment

Method	URI	Description
PUT	/v1/{tenant_id}/software_deployments/{deployment_id}	Updates a software deployment.

Normal response codes: 200

16.9.8.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{deployment_id}	String	The deployment ID.

Example 16.44. Update deployment: JSON request

```
{
    "status": "COMPLETE",
    "output_values": {
        "deploy_stdout": "Writing to /tmp/baaaaaa\\nWritten to /tmp/baaaaaa\\n",
        "deploy_stderr": "+ echo Writing to /tmp/baaaaaa\\n+ echo fooooo\\n+
cat /tmp/baaaaaa\\n+ echo -n The file /tmp/baaaaaa contains fooooo for server
ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE\\n+ echo Written to /tmp/
baaaaaa\\n+ echo Output to stderr\\nOutput to stderr\\n",
        "deploy_status_code": 0,
        "result": "The file /tmp/baaaaaa contains fooooo for server
ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE"
    },
    "status_reason": "Outputs received"
}
```

16.9.8.2. Response

Example 16.45. Update deployment: JSON response

```
{
    "software_deployment": {
        "status": "COMPLETE",
        "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
        "config_id": "3d5ec2a8-7004-43b6-a7f6-542bdbe9d434",
        "output_values": {
            "deploy_stdout": "Writing to /tmp/baaaaaa\\nWritten to /tmp/baaaaaa\\n",
            "deploy_stderr": "+ echo Writing to /tmp/baaaaaa\\n+ echo fooooo\\n+
cat /tmp/baaaaaa\\n+ echo -n The file /tmp/baaaaaa contains fooooo for server
ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE\\n+ echo Written to /tmp/
baaaaaa\\n+ echo Output to stderr\\nOutput to stderr\\n",
            "deploy_status_code": 0,
            "result": "The file /tmp/baaaaaa 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"
    }
```

16.9.9. Delete deployment

Method	URI	Description
DELETE	/v1/{tenant_id}/software_deployments/{deployment_id}	Deletes a software deployment.

Normal response codes: 204

16.9.9.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.
{deployment_id}	String	The deployment ID.

This operation does not accept a request body.

16.10. Manage service

Method	URI	Description
GET	/v1/{tenant_id}/services	Enables administrative users to view details for all orchestration engines.

16.10.1. 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

16.10.1.1. Request

This table shows the URI parameters for the show orchestration engine status request:

Name	Type	Description
{tenant_id}	String	The ID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

16.10.1.2. Response

Example 16.46. 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"
    }
]
```

17. 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: 2.6`". 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.

Method	URI	Description
API versions		
GET	/	Lists all Shared File Systems API versions.
API extensions		
GET	/v2/{tenant_id}/extensions	Lists extensions.
Limits		
GET	/v2/{tenant_id}/limits	Gets 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,is_public,consistency_group_id}	Lists 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,is_public,consistency_group_id}	Lists shares with details.
GET	/v2/{tenant_id}/shares/{share_id}	Shows information about 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 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.

Method	URI	Description
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.
Share snapshots		
POST	/v2/{tenant_id}/snapshots	Creates a snapshot from a share.
GET	/v2/{tenant_id}/snapshots	Lists all share snapshots.
GET	/v2/{tenant_id}/snapshots/detail	Lists all share snapshots with details.
GET	/v2/{tenant_id}/snapshots/{snapshot_id}	Shows information 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	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.
GET	/v2/{tenant_id}/share-networks/detail	Lists all share networks with details.
GET	/v2/{tenant_id}/share-networks/{share_network_id}	Shows information 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	Add security service to a share network.
POST	/v2/{tenant_id}/share-networks/{share_network_id}/action	Remove 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.

Method	URI	Description
GET	/v2/{tenant_id}/security-services/{security_service_id}	Shows information 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 information for a share server.
DELETE	/v2/{tenant_id}/share-servers/{share_server_id}	Deletes a share server.
GET	/v2/{tenant_id}/share-servers/{share_server_id}/details	Shows the details for 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 information about 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 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.
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}/os-share-type-access	Shows access information for 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.
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}/os-services	Lists services.
PUT	/v2/{tenant_id}/os-services/enable	Enables a service.
PUT	/v2/{tenant_id}/os-services/disable	Disables a service.

Method	URI	Description
Availability zones		
GET	/v2/{tenant_id}/os-availability-zone	Lists all availability zones.
Manage share		
POST	/v2/{tenant_id}/os-share-manage	Configures Shared File Systems to manage a share.
Unmanage 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}/os-quota-sets/{tenant_id}{?user_id}	Shows quotas for a tenant.
PUT	/v2/{tenant_id}/os-quota-sets/{tenant_id}{?user_id}	Updates quotas for a tenant.
DELETE	/v2/{tenant_id}/os-quota-sets/{tenant_id}{?user_id}	Deletes quotas for a tenant. The quota will revert back to default.
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}/defaults	Shows default quotas for a tenant.

17.1. API versions

Lists information for all Shared File Systems API versions.

Method	URI	Description
GET	/	Lists all Shared File Systems API versions.

17.1.1. List versions

Method	URI	Description
GET	/	Lists all Shared File Systems API versions.

Normal response codes: 200

17.1.1.1. Request

This table shows the header parameters for the list versions request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". If you omit this header, the default micro-version is 2.0.

This operation does not accept a request body.

17.1.1.2. Response

Example 17.1. List versions: JSON response

```
{
    "versions": [
        {
            "status": "SUPPORTED",
            "updated": "2015-08-27T11:33:21Z",
            "links": [
                {
                    "href": "http://docs.openstack.org/",
                    "type": "text/html",
                    "rel": "describedby"
                },
                {
                    "href": "http://172.18.198.54:8786/v1/",
                    "rel": "self"
                }
            ],
            "min_version": "",
            "version": "",
            "media-types": [
                {
                    "base": "application/json",
                    "type": "application/vnd.openstack.share+json;version=1"
                }
            ],
            "id": "v1.0"
        },
        {
            "status": "CURRENT",
            "updated": "2015-08-27T11:33:21Z",
            "links": [
                {
                    "href": "http://docs.openstack.org/",
                    "type": "text/html",

```

```
        "rel": "describedby"
    },
{
    "href": "http://172.18.198.54:8786/v2/" ,
    "rel": "self"
}
],
"min_version": "2.0",
"version": "2.6",
"media-types": [
    {
        "base": "application/json",
        "type": "application/vnd.openstack.share+json;version=1"
    }
],
"id": "v2.0"
}
]
```

17.2. API extensions

Lists available Shared File Systems API extensions.

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

17.2.1. List extensions

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

Normal response codes: 200

17.2.1.1. Request

This table shows the header parameters for the list extensions request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.2.1.2. Response

Example 17.2. 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."
    }
]
```

17.3. 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	/v2/{tenant_id}/limits	Gets share limits.

17.3.1. Get share limits

Method	URI	Description
GET	/v2/{tenant_id}/limits	Gets share limits.

Normal response codes: 200

17.3.1.1. Request

This table shows the header parameters for the get share limits request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the get share limits request:

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

This operation does not accept a request body.

17.3.1.2. Response

Example 17.3. Get 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
        }
    }
}
```

17.4. 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).

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 17.1. 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, is_public, consistency_group_id}	Lists 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, is_public, consistency_group_id}	Lists shares with details.
GET	/v2/{tenant_id}/shares/{share_id}	Shows information about a share.

Method	URI	Description
PUT	/v2/{tenant_id}/shares/{share_id}	Updates a share.
DELETE	/v2/{tenant_id}/shares/{share_id} {?consistency_group_id}	Deletes a share.

17.4.1. Create share

Method	URI	Description
POST	/v2/{tenant_id}/shares	Creates a share.

Normal response codes: 200

17.4.1.1. Request

This table shows the header parameters for the create share request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

Example 17.4. 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"  
    }  
  }  
}
```

17.4.1.2. Response

Example 17.5. 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,
        "task_state": null,
        "snapshot_support": true,
        "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
        "source_cgssnapshot_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"
    }
}
```

17.4.2. 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,is_public,consistency_group_id}	Lists shares.

Normal response codes: 200

17.4.2.1. Request

This table shows the header parameters for the list shares request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

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

Name	Type	Description
all_tenants	Bool <i>(Optional)</i>	(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 <i>(Optional)</i>	The share name.
status	String <i>(Optional)</i>	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 <i>(Optional)</i>	The ID of the share server.
metadata	String <i>(Optional)</i>	One or more metadata key-value pairs, as a dictionary of strings.
extra_specs	String <i>(Optional)</i>	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 <i>(Optional)</i>	(Since API v2.6.) The ID of the share type.

Name	Type	Description
limit <i>(Optional)</i>	Int	The maximum number of shares to return.
offset <i>(Optional)</i>	Int	The offset to define start point of share listing.
sort_key <i>(Optional)</i>	String	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 <i>(Optional)</i>	String	The direction to sort a list of shares. A valid value is asc, or desc.
snapshot_id <i>(Optional)</i>	UUID	The snapshot ID, that was used for the share.
host <i>(Optional)</i>	String	The share host name.
share_network_id <i>(Optional)</i>	UUID	The share network ID.
project_id <i>(Optional)</i>	UUID	The ID of the project in which the share was created. Useful with all_tenants parameter.
is_public <i>(Optional)</i>	Bool	The level of visibility for the share. Set to true to list only public shares. Set to false to list only private shares.
consistency_group_id <i>(Optional)</i>	UUID	(Since API v2.4.) The ID of a consistency group in which the share was created.

This operation does not accept a request body.

17.4.2.2. Response

Example 17.6. 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/
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"
    }
]
```

17.4.3. List shares (detailed)

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,is_public,consistency_group_id}	Lists shares with details.

Normal response codes: 200

17.4.3.1. Request

This table shows the header parameters for the list shares (detailed) request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list shares (detailed) request:

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

This table shows the query parameters for the list shares (detailed) request:

Name	Type	Description
all_tenants	Bool <i>(Optional)</i>	(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 <i>(Optional)</i>	The share name.
status	String <i>(Optional)</i>	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 <i>(Optional)</i>	The ID of the share server.
metadata	String <i>(Optional)</i>	One or more metadata key-value pairs, as a dictionary of strings.
extra_specs	String <i>(Optional)</i>	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 <i>(Optional)</i>	(Since API v2.6.) The ID of the share type.
limit	Int <i>(Optional)</i>	The maximum number of shares to return.

Name	Type	Description
offset	Int <i>(Optional)</i>	The offset to define start point of share listing.
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 snapshot ID, that was used for the share.
host	String <i>(Optional)</i>	The share host name.
share_network_id	UUID <i>(Optional)</i>	The share network ID.
project_id	UUID <i>(Optional)</i>	The ID of the project in which the share was created. Useful with all_tenants parameter.
is_public	Bool <i>(Optional)</i>	The level of visibility for the share. Set to true to list only public shares. Set to false to list only private shares.
consistency_group_id	UUID <i>(Optional)</i>	(Since API v2.4.) The ID of a consistency group in which the share was created.

This operation does not accept a request body.

17.4.3.2. Response

Example 17.7. List shares (detailed): 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/v2/16e1ab15c35a457e9c2b2aa189f544e1/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,
      "status": "available"
    }
  ]
}
```

```
        "share_type": "25747776-08e5-494f-ab40-a64b9d20d8f7",
        "share_type_name": "default",
        "export_location": null,
        "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
        "project_id": "16elab15c35a457e9c2b2aa189f544e1",
        "metadata": {},
        "status": "error",
        "description": "There is a share description.",
        "host": "manila2@generic1#GENERIC1",
        "task_state": null,
        "is_public": true,
        "snapshot_support": true,
        "name": "my_share4",
        "created_at": "2015-09-16T18:19:50.000000",
        "share_proto": "NFS",
        "volume_type": "default",
        "source_cgssnapshot_member_id": null
    },
    {
        "links": [
            {
                "href": "http://172.18.198.54:8786/v2/
16elab15c35a457e9c2b2aa189f544e1/shares/c4a2ced4-2c9f-4ael-adaa-6171833e64df",
                "rel": "self"
            },
            {
                "href": "http://172.18.198.54:8786/
16elab15c35a457e9c2b2aa189f544e1/shares/c4a2ced4-2c9f-4ael-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-4ael-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": "16elab15c35a457e9c2b2aa189f544e1",
        "metadata": {},
        "status": "available",
        "description": "Changed description.",
        "host": "manila2@generic1#GENERIC1",
        "task_state": null,
        "is_public": true,
        "snapshot_support": true,
        "name": "my_share4",
        "created_at": "2015-09-16T17:26:28.000000",
        "share_proto": "NFS",
        "volume_type": "default",
        "source_cgssnapshot_member_id": null
    }
}
```

```
    ]  
}
```

17.4.4. Show share

Method	URI	Description
GET	/v2/{tenant_id}/shares/{share_id}	Shows information about a share.

Normal response codes: 200

17.4.4.1. Request

This table shows the header parameters for the show share request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". If you omit this header, the default micro-version is 2.0.

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

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

This operation does not accept a request body.

17.4.4.2. Response

Example 17.8. Show 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",
        "status": "available"
    }
}
```

```
    "metadata": {
        "project": "my_app",
        "aim": "doc"
    },
    "status": "available",
    "description": "My custom share London",
    "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_cgssnapshot_member_id": null
}
}
```

17.4.5. 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

17.4.5.1. 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: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

Example 17.9. Update share: JSON request

```
{
  "share": {
    "is_public": true,
    "display_description": "Changing the share description."
  }
}
```

17.4.5.2. Response

Example 17.10. 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_cgssnapshot_member_id": null  
}  
}
```

17.4.6. Delete share

Method	URI	Description
DELETE	/v2/{tenant_id}/shares/{share_id}{?consistency_group_id}	Deletes a share.

Normal response codes: 202

17.4.6.1. Request

This table shows the header parameters for the delete share request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

This table shows the query parameters for the delete share request:

Name	Type	Description
consistency_group_id	UUID <i>(Optional)</i>	(Since API v2.4.) The ID of a consistency group in which 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.

17.5. 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.
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.

17.5.1. Show share metadata

Method	URI	Description
GET	/v2/{tenant_id}/shares/{share_id}/metadata	Shows the metadata for a share.

Normal response codes: 200

17.5.1.1. Request

This table shows the header parameters for the show share metadata request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

This operation does not accept a request body.

17.5.1.2. Response

Example 17.11. Show share metadata: JSON response

```
{
  "metadata": {
    "project": "my_app",
    "aim": "doc"
  }
}
```

17.5.2. Update share metadata

Method	URI	Description
PUT	/v2/{tenant_id}/shares/{share_id}/metadata	Updates the metadata for a share.

Normal response codes: 200

17.5.2.1. Request

This table shows the header parameters for the update share metadata request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

Example 17.12. Update share metadata: JSON request

```
{
  "metadata": {
    "aim": "changed_doc",
    "project": "my_app",
    "new_metadata_key": "new_information"
  }
}
```

17.5.2.2. Response

Example 17.13. Update share metadata: JSON response

```
{
  "metadata": {
    "aim": "changed_doc",
    "project": "my_app",
    "new_metadata_key": "new_information"
  }
}
```

17.5.3. Set share metadata

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/metadata	Sets the metadata on a share.

Normal response codes: 200

17.5.3.1. Request

This table shows the header parameters for the set share metadata request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

Example 17.14. Set share metadata: JSON request

```
{
    "metadata": {
        "key1": "value1"
    }
}
```

17.5.3.2. Response

Example 17.15. Set share metadata: JSON response

```
{
    "metadata": {
        "aim": "changed_doc",
        "project": "my_app",
        "key1": "value1",
        "new_metadata_key": "new_information",
        "key": "value"
    }
}
```

17.5.4. 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

17.5.4.1. Request

This table shows the header parameters for the unset share metadata request:

Name	Type	Description
x-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.
{metadata_key}	String	The metadata key.

This operation does not accept a request body.

17.6. 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`

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.

17.6.1. Grant access

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Grants access to a share.

Normal response codes: 200

17.6.1.1. Request

This table shows the header parameters for the grant access request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

Example 17.16. Grant access: JSON request

```
{
    "os-allow_access": {
        "access_level": "rw",
        "access_type": "ip",
        "access_to": "0.0.0.0/0"
    }
}
```

17.6.1.2. Response

Example 17.17. 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"
    }
}
```

17.6.2. Revoke access

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Revokes access from a share.

Normal response codes: 202

17.6.2.1. Request

This table shows the header parameters for the revoke access request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

Example 17.18. Revoke access: JSON request

```
{
    "os-deny_access": {
        "access_id": "a25b2df3-90bd-4add-afa6-5f0dbbd50452"
    }
}
```

17.6.3. List access rules

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Lists access rules for a share.

Normal response codes: 200

17.6.3.1. Request

This table shows the header parameters for the list access rules request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

Example 17.19. List access rules: JSON request

```
{
    "os-access_list": null
}
```

17.6.3.2. Response

Example 17.20. 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"
        }
    ]
}
```

17.6.4. 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

17.6.4.1. Request

This table shows the header parameters for the reset share state request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

Example 17.21. Reset share state: JSON request

```
{
    "os-reset_status": {
        "status": "error"
    }
}
```

17.6.5. 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

17.6.5.1. Request

This table shows the header parameters for the force-delete share request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

Example 17.22. Force-delete share: JSON request

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

17.6.6. Extend share

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Increases the size of a share.

Normal response codes: 202

17.6.6.1. Request

This table shows the header parameters for the extend share request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

Example 17.23. Extend share: JSON request

```
{
    "os-extend": {
        "new_size": 2
    }
}
```

17.6.7. Shrink share

Method	URI	Description
POST	/v2/{tenant_id}/shares/{share_id}/action	Shrinks the size of a share.

Normal response codes: 202

17.6.7.1. Request

This table shows the header parameters for the shrink share request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of the share.

Example 17.24. Shrink share: JSON request

```
{
    "os-shrink": {
        "new_size": 1
    }
}
```

17.7. 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, update, and delete share snapshots. After you create 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

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	/v2/{tenant_id}/snapshots	Creates a snapshot from a share.
GET	/v2/{tenant_id}/snapshots	Lists all share snapshots.
GET	/v2/{tenant_id}/snapshots/detail	Lists all share snapshots with details.
GET	/v2/{tenant_id}/snapshots/{snapshot_id}	Shows information 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	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.

17.7.1. Create share snapshot

Method	URI	Description
POST	/v2/{tenant_id}/snapshots	Creates a snapshot from a share.

Normal response codes: 200

17.7.1.1. Request

This table shows the header parameters for the create share snapshot request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

Example 17.25. 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"
    }
}
```

17.7.1.2. Response

Example 17.26. 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://172.18.198.54:8786/16e1ab15c35a457e9c2b2aa189f544e1/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
    }
}
```

17.7.2. List share snapshots

Method	URI	Description
GET	/v2/{tenant_id}/snapshots	Lists all share snapshots.

Normal response codes: 200

17.7.2.1. Request

This table shows the header parameters for the list share snapshots request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.7.2.2. Response

Example 17.27. 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://172.18.198.54:8786/16e1ab15c35a457e9c2b2aa189f544e1/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"
}
]
```

17.7.3. List share snapshots (detailed)

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

Normal response codes: 200

17.7.3.1. Request

This table shows the header parameters for the list share snapshots (detailed) request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the list share snapshots (detailed) request:

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

This operation does not accept a request body.

17.7.3.2. Response

Example 17.28. List share snapshots (detailed): 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/086alaaa6-c425-4ecd-9612-391a3b1b9375",
                    "rel": "self"
                },
                {
                    "href": "http://172.18.198.54:8786/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/086alaaa6-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": "086alaaa6-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
    }
]
```

17.7.4. Show share snapshot details

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

Normal response codes: 200

17.7.4.1. 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: 2.6". 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 tenant ID in a multi-tenancy cloud.
{snapshot_id}	UUID	The ID of the snapshot.

This operation does not accept a request body.

17.7.4.2. Response

Example 17.29. 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://172.18.198.54:8786/16e1ab15c35a457e9c2b2aa189f544e1/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
  }
}
```

```
    }  
}
```

17.7.5. 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

17.7.5.1. 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: 2.6". 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 tenant ID in a multi-tenancy cloud.
{snapshot_id}	UUID	The ID of the snapshot.

Example 17.30. 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"
  }
}
```

17.7.5.2. Response

Example 17.31. 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  
}  
}
```

17.7.6. Delete share snapshot

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

Normal response codes: 202

17.7.6.1. Request

This table shows the header parameters for the delete share snapshot request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{snapshot_id}	UUID	The ID of the snapshot.

This operation does not accept a request body.

17.7.7. 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

17.7.7.1. Request

This table shows the header parameters for the reset share snapshot state request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{snapshot_id}	UUID	The ID of the snapshot.

Example 17.32. Reset share snapshot state: JSON request

```
{
    "os-reset_status": {
        "status": "error"
    }
}
```

17.7.8. 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

17.7.8.1. Request

This table shows the header parameters for the force-delete share snapshot request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{snapshot_id}	UUID	The ID of the snapshot.

Example 17.33. Force-delete share snapshot: JSON request

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

17.8. 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 information 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	Add security service to a share network.
POST	/v2/{tenant_id}/share-networks/{share_network_id}/action	Remove security service from a share network.

17.8.1. Create share network

Method	URI	Description
POST	/v2/{tenant_id}/share-networks	Creates a share network.

Normal response codes: 200

17.8.1.1. Request

This table shows the header parameters for the create share network request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

Example 17.34. Create share network: JSON request

```
{
  "share_network": {
    "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5calf2109",
    "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
    "name": "my_network",
    "description": "This is my share network"
  }
}
```

17.8.1.2. Response

Example 17.35. 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-4d36-8b95-67b5calf2109",
    "ip_version": null,
    "nova_net_id": null,
    "cidr": null,
    "project_id": "e10a683c20da41248cf5e1ab3d88c62",
    "network_type": null,
    "description": "This is my share network"
  }
}
```

17.8.2. List share networks

Method	URI	Description
GET	/v2/{tenant_id}/share-networks	Lists all share networks.

Normal response codes: 200

17.8.2.1. Request

This table shows the header parameters for the list share networks request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.8.2.2. Response

Example 17.36. 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
    }
  ]
}
```

17.8.3. 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

17.8.3.1. Request

This table shows the header parameters for the list share networks with details request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.8.3.2. Response

Example 17.37. 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-4d36-8b95-67b5calf2109",
      "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-4d36-8b95-67b5calf2109",
      "ip_version": null,
      "nova_net_id": null,
    }
  ]
}
```

```
        "cidr": null,
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "network_type": null,
        "description": "desechr"
    },
    {
        "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
    }
]
```

17.8.4. Show share network

Method	URI	Description
GET	/v2/{tenant_id}/share-networks/{share_network_id}	Shows information for a share network.

Normal response codes: 200

17.8.4.1. Request

This table shows the header parameters for the show share network request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show share network request:

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

This operation does not accept a request body.

17.8.4.2. Response

Example 17.38. Show share network: 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"
  }
}
```

17.8.5. 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

17.8.5.1. Request

This table shows the header parameters for the update share network request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_network_id}	UUID	The ID of the share network.

Example 17.39. Update share network: JSON request

```
{
  "share_network": {
    "description": "i'm adding a description"
  }
}
```

17.8.5.2. Response

Example 17.40. 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-4d36-8b95-67b5calf2109",
    "ip_version": "4",
    "nova_net_id": null,
    "cidr": null,
    "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "network_type": null,
    "description": "i'm adding a description"
  }
}
```

}

17.8.6. Delete share network

Method	URI	Description
DELETE	/v2/{tenant_id}/share-networks/{share_network_id}	Deletes a share network.

Normal response codes: 202

17.8.6.1. Request

This table shows the header parameters for the delete share network request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_network_id}	UUID	The ID of the share network.

This operation does not accept a request body.

17.8.7. Add security service to share network

Method	URI	Description
POST	/v2/{tenant_id}/share-networks/{share_network_id}/action	Add security service to a share network.

Normal response codes: 200

17.8.7.1. Request

This table shows the header parameters for the add security service to share network request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_network_id}	UUID	The ID of the share network.

Example 17.41. Add security service to share network: JSON request

```
{
    "add_security_service": {
        "security_service_id": "3c829734-0679-4c17-9637-801da48c0d5f"
    }
}
```

17.8.7.2. Response

Example 17.42. 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
    }
}
```

17.8.8. Remove security service from share network

Method	URI	Description
POST	/v2/{tenant_id}/share-networks/{share_network_id}/action	Remove security service from a share network.

Normal response codes: 200

17.8.8.1. Request

This table shows the header parameters for the remove security service from share network request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_network_id}	UUID	The ID of the share network.

Example 17.43. Remove security service from share network: JSON request

```
{
    "remove_security_service": {
        "security_service_id": "3c829734-0679-4c17-9637-801da48c0d5f"
    }
}
```

17.8.8.2. Response

Example 17.44. 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
    }
}
```

17.9. 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	/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 information 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.

17.9.1. Create security service

Method	URI	Description
POST	/v2/{tenant_id}/security-services	Creates a security service.

Normal response codes: 200

17.9.1.1. Request

This table shows the header parameters for the create security service request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

Example 17.45. 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"
    }
}
```

17.9.1.2. Response

Example 17.46. 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"
    }
}
```

}

17.9.2. List security services

Method	URI	Description
GET	/v2/{tenant_id}/security-services	Lists all security services.

Normal response codes: 200

17.9.2.1. Request

This table shows the header parameters for the list security services request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.9.2.2. Response

Example 17.47. 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"
    }
  ]
}
```

17.9.3. 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

17.9.3.1. Request

This table shows the header parameters for the list security services with details request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.9.3.2. Response

Example 17.48. 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": "10.0.0.1/24",
            "user": "root",
            "password": "supersecret",
            "type": "kerberos",
            "id": "3c829734-0679-4c17-9637-801da48c0d5f"
        }
    ]
}
```

```
        "dns_ip": "10.0.0.0/24",
        "user": null,
        "password": null,
        "type": "ldap",
        "id": "5a1d3a12-34a7-4087-8983-50e9ed03509a",
        "share_networks": []
    }
]
```

17.9.4. Show security service

Method	URI	Description
GET	/v2/{tenant_id}/security-services/{security_service_id}	Shows information for a security service.

Normal response codes: 200

17.9.4.1. Request

This table shows the header parameters for the show security service request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show security service request:

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

This operation does not accept a request body.

17.9.4.2. Response

Example 17.49. Show security service: 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"
  }
}
```

17.9.5. 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

17.9.5.1. Request

This table shows the header parameters for the update security service request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{security_service_id}	UUID	The ID of the security service.

Example 17.50. Update security service: JSON request

```
{
    "security_service": {
        "domain": "my_domain",
        "password": "*****",
        "user": "new_user",
        "description": "Adding a description"
    }
}
```

17.9.5.2. Response

Example 17.51. 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",
        "share_network": null
    }
}
```

```
        "type": "kerberos",
        "id": "3c829734-0679-4c17-9637-801da48c0d5f",
        "description": "Adding a description"
    }
}
```

17.9.6. Delete security service

Method	URI	Description
DELETE	/v2/{tenant_id}/security-services/{security_service_id}	Deletes a security service.

Normal response codes: 202

17.9.6.1. Request

This table shows the header parameters for the delete security service request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{security_service_id}	UUID	The ID of the security service.

This operation does not accept a request body.

17.10. 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 17.2. 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 information for a share server.
DELETE	/v2/{tenant_id}/share-servers/{share_server_id}	Deletes a share server.
GET	/v2/{tenant_id}/share-servers/{share_server_id}/details	Shows the details for a share server.

17.10.1. List share servers

Method	URI	Description
GET	/v2/{tenant_id}/share-servers	Lists all share servers.

Normal response codes: 200

17.10.1.1. Request

This table shows the header parameters for the list share servers request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.10.1.2. Response

Example 17.52. 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-a8dfbaa3cc73"
    }
  ]
}
```

17.10.2. Show share server

Method	URI	Description
GET	/v2/{tenant_id}/share-servers/{share_server_id}	Shows information for a share server.

Normal response codes: 200

17.10.2.1. Request

This table shows the header parameters for the show share server request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show share server request:

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

This operation does not accept a request body.

17.10.2.2. Response

Example 17.53. Show share server: 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"
  }
}
```

17.10.3. 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

17.10.3.1. Request

This table shows the header parameters for the delete share server request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_server_id}	UUID	The ID of the share server.

This operation does not accept a request body.

17.10.4. Show share server details

Method	URI	Description
GET	/v2/{tenant_id}/share-servers/{share_server_id}/details	Shows the details for a share server.

Normal response codes: 200

17.10.4.1. Request

This table shows the header parameters for the show share server details request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_server_id}	UUID	The ID of the share server.

This operation does not accept a request body.

17.10.4.2. Response

Example 17.54. Show share server details: JSON response

```
{
  "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"
  }
}
```

17.11. 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 information about a share instance.

Method	URI	Description
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.

17.11.1. List share instances

Method	URI	Description
GET	/v2/{tenant_id}/share_instances	Lists all share instances.

Normal response codes: 200

17.11.1.1. Request

This table shows the header parameters for the list share instances request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.11.1.2. Response

Example 17.55. List share instances: JSON response

```
{
  "share_instances": [
    {
      "status": "error",
      "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
      "availability_zone": "nova",
      "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",
      "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"
}
]
```

17.11.2. Show share instance

Method	URI	Description
GET	/v2/{tenant_id}/share_instances/{share_instance_id}	Shows information about a share instance.

Normal response codes: 200

17.11.2.1. Request

This table shows the header parameters for the show share instance request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show share instance request:

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

This operation does not accept a request body.

17.11.2.2. Response

Example 17.56. Show share instance: JSON response

```
{
  "share_instance": {
    "status": "available",
    "share_id": "d94a8548-2079-4be0-b21c-0a887acd31ca",
    "availability_zone": "nova",
    "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"
  }
}
```

17.11.3. 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

17.11.3.1. Request

This table shows the header parameters for the reset share instance state request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_instance_id}	UUID	The ID of the share instance.

Example 17.57. Reset share instance state: JSON request

```
{
    "os-reset_status": {
        "status": "available"
    }
}
```

17.11.4. 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

17.11.4.1. 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: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_instance_id}	UUID	The ID of the share instance.

Example 17.58. Force-delete share instance: JSON request

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

17.12. 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 `os-share-type-access:is_public` 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	<code>/v2/{tenant_id}/types</code>	Creates a share type.
GET	<code>/v2/{tenant_id}/types</code>	Lists all share types.
GET	<code>/v2/{tenant_id}/types/default</code>	Lists default share types.
DELETE	<code>/v2/{tenant_id}/types/{share_type_id}</code>	Deletes a share type.
POST	<code>/v2/{tenant_id}/types/{share_type_id}/action</code>	Adds share type access for a project.
POST	<code>/v2/{tenant_id}/types/{share_type_id}/action</code>	Removes share type access from a project.
GET	<code>/v2/{tenant_id}/types/{share_type_id}/os-share-type-access</code>	Shows access information for 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.

17.12.1. Create share type

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

Normal response codes: 200

17.12.1.1. Request

This table shows the header parameters for the create share type request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

Example 17.59. 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"
    }
}
```

17.12.1.2. Response

Example 17.60. 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"
}
}
```

17.12.2. List share types

Method	URI	Description
GET	/v2/{tenant_id}/types	Lists all share types.

Normal response codes: 200

17.12.2.1. Request

This table shows the header parameters for the list share types request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.12.2.2. Response

Example 17.61. 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"  
}  
]  
}
```

17.12.3. List default share types

Method	URI	Description
GET	/v2/{tenant_id}/types/default	Lists default share types.

Normal response codes: 200

17.12.3.1. Request

This table shows the header parameters for the list default share types request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.12.3.2. Response

Example 17.62. 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"
    }
}
```

17.12.4. Delete share type

Method	URI	Description
DELETE	/v2/{tenant_id}/types/{share_type_id}	Deletes a share type.

Normal response codes: 202

17.12.4.1. Request

This table shows the header parameters for the delete share type request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_type_id}	UUID	The ID of the share type.

This operation does not accept a request body.

17.12.5. 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

17.12.5.1. Request

This table shows the header parameters for the add share type access request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_type_id}	UUID	The ID of the share type.

Example 17.63. Add share type access: JSON request

```
{
    "addProjectAccess": {
        "project": "e1284adea3ee4d2482af5ed214f3ad90"
    }
}
```

17.12.6. 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

17.12.6.1. Request

This table shows the header parameters for the remove share type access request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_type_id}	UUID	The ID of the share type.

Example 17.64. Remove share type access: JSON request

```
{
    "removeProjectAccess": {
        "project": "818a3f48dcd644909b3fa2e45a399a27"
    }
}
```

17.12.7. Show share type access

Method	URI	Description
GET	/v2/{tenant_id}/types/{share_type_id}/os-share-type-access	Shows access information for a share type.

You can view access information for private share types only.

Normal response codes: 200

17.12.7.1. Request

This table shows the header parameters for the show share type access request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the show share type access request:

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

This operation does not accept a request body.

17.12.7.2. Response

Example 17.65. Show share type access: JSON response

```
{
  "share_type_access": [
    {
      "share_type_id": "1732f284-401d-41d9-a494-425451e8b4b8",
      "project_id": "818a3f48dc644909b3fa2e45a399a27"
    },
    {
      "share_type_id": "1732f284-401d-41d9-a494-425451e8b4b8",
      "project_id": "e1284adea3ee4d2482af5ed214f3ad90"
    }
  ]
}
```

17.12.8. 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

17.12.8.1. Request

This table shows the header parameters for the list extra specs request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_type_id}	UUID	The ID of the share type.

This operation does not accept a request body.

17.12.8.2. Response

Example 17.66. List extra specs: JSON response

```
{
    "extra_specs": {
        "snapshot_support": "True",
        "driver_handles_share_servers": "True"
    }
}
```

17.12.9. Set extra spec

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

17.12.9.1. Request

This table shows the header parameters for the set extra spec request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". If you omit this header, the default micro-version is 2.0.

This table shows the URI parameters for the set extra spec request:

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

Example 17.67. Set extra spec: JSON request

```
{
    "extra_specs": {
        "my_key": "my_value"
    }
}
```

17.12.9.2. Response

Example 17.68. Set extra spec: JSON response

```
{
    "extra_specs": {
        "my_key": "my_value"
    }
}
```

17.12.10. 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

17.12.10.1. Request

This table shows the header parameters for the unset an extra spec request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_type_id}	UUID	The ID of the share type.
{extra-spec-key}	String	The extra specification key.

This operation does not accept a request body.

17.13. 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.
GET	/v2/{tenant_id}/scheduler-stats/pools/detail	Lists all storage pools for a back end, with details.

17.13.1. 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

17.13.1.1. Request

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

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.13.1.2. Response

Example 17.69. 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"
    }
  ]
}
```

17.13.2. 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

17.13.2.1. 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 <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.13.2.2. Response

Example 17.70. List back-end storage pools with details: JSON response

```
{
  "pools": [
    {
      "pool": "LONDON",
      "host": "nosb-devstack",
      "name": "nosb-devstack@london#LONDON",
      "capabilities": {
        "QoS_support": 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"
    ]
}
```

17.14. 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.

Method	URI	Description
GET	/v2/{tenant_id}/os-services	Lists services.
PUT	/v2/{tenant_id}/os-services/enable	Enables a service.
PUT	/v2/{tenant_id}/os-services/disable	Disables a service.

17.14.1. List services

Method	URI	Description
GET	/v2/{tenant_id}/os-services	Lists services.

Normal response codes: 200

17.14.1.1. Request

This table shows the header parameters for the list services request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.14.1.2. Response

Example 17.71. 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
    }
  ]
}
```

17.14.2. Enable service

Method	URI	Description
PUT	/v2/{tenant_id}/os-services/enable	Enables a service.

Normal response codes: 200

17.14.2.1. Request

This table shows the header parameters for the enable service request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

Example 17.72. Enable service: JSON request

```
{
    "binary": "manila-share",
    "host": "openstack@cmode"
}
```

17.14.2.2. Response

Example 17.73. Enable service: JSON response

```
{
    "disabled": false,
    "binary": "manila-share",
    "host": "openstack@cmode"
}
```

17.14.3. Disable service

Method	URI	Description
PUT	/v2/{tenant_id}/os-services/disable	Disables a service.

Normal response codes: 200

17.14.3.1. Request

This table shows the header parameters for the disable service request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

Example 17.74. Disable service: JSON request

```
{
    "binary": "manila-share",
    "host": "openstack@cmode"
}
```

17.14.3.2. Response

Example 17.75. Disable service: JSON response

```
{
    "disabled": true,
    "binary": "manila-share",
    "host": "openstack@cmode"
}
```

17.15. Availability zones

Describes availability zones.

Method	URI	Description
GET	/v2/{tenant_id}/os-availability-zone	Lists all availability zones.

17.15.1. List availability zones

Method	URI	Description
GET	/v2/{tenant_id}/os-availability-zone	Lists all availability zones.

Normal response codes: 200

17.15.1.1. Request

This table shows the header parameters for the list availability zones request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

This operation does not accept a request body.

17.15.1.2. Response

Example 17.76. 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"
        }
    ]
}
```

17.16. Manage share

Configures Shared File Systems to manage a share.

Method	URI	Description
POST	/v2/{tenant_id}/os-share-manage	Configures Shared File Systems to manage a share.

17.16.1. Manage share

Method	URI	Description
POST	/v2/{tenant_id}/os-share-manage	Configures Shared File Systems to manage a share.

Normal response codes: 200

17.16.1.1. Request

This table shows the header parameters for the manage share request:

Name	Type	Description
X-Openstack-Manila-Api-Version <i>(Optional)</i>	String	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.

Example 17.77. Manage share: JSON request

```
{
  "share": {
    "protocol": "nfs",
    "name": "share_texas1",
    "share_type": "d",
    "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."
  }
}
```

17.16.1.2. Response

Example 17.78. 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"
      }
    ]
  }
}
```

```
        "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": "16elab15c35a457e9c2b2aa189f544e1",
"metadata": {},
"status": "manage_starting",
"description": "Lets manage share.",
"host": "manila2@unmanage1#UNMANAGE1",
"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
}
}
```

17.17. Unmanage share

Configures Shared File Systems to 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.

Method	URI	Description
POST	/v2/{tenant_id}/os-share-unman- age/{share_id}/unmanage	Configures Shared File Systems to stop managing a share.

17.17.1. 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

17.17.1.1. Request

This table shows the header parameters for the unmanage share request:

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{share_id}	UUID	The ID of a share.

This operation does not accept a request body.

17.18. Quota sets

Provides quotas management support.

Method	URI	Description
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}{?user_id}	Shows quotas for a tenant.
PUT	/v2/{tenant_id}/os-quota-sets/{tenant_id}{?user_id}	Updates quotas for a tenant.
DELETE	/v2/{tenant_id}/os-quota-sets/{tenant_id}{?user_id}	Deletes quotas for a tenant. The quota will revert back to default.
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}/defaults	Shows default quotas for a tenant.

17.18.1. Show quotas

Method	URI	Description
GET	/v2/{tenant_id}/os-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

17.18.1.1. Request

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

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{tenant_id}	UUID	The ID 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 admin tenant.

This operation does not accept a request body.

17.18.1.2. Response

Example 17.79. Show quotas: JSON response

```
{
  "quota_set": {
    "gigabytes": 1000,
    "shares": 50,
    "snapshot_gigabytes": 1000,
    "snapshots": 50,
    "id": "16e1ab15c35a457e9c2b2aa189f544e1",
    "share_networks": 10
  }
}
```

17.18.2. Update quotas

Method	URI	Description
PUT	/v2/{tenant_id}/os-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

17.18.2.1. Request

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

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{tenant_id}	UUID	The ID 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 admin tenant.

Example 17.80. Update quotas: JSON request

```
{
    "quota_set": {
        "tenant_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "snapshot_gigabytes": 999,
        "snapshots": 49,
        "share_networks": 9
    }
}
```

17.18.2.2. Response

Example 17.81. Update quotas: JSON response

```
{
    "quota_set": {
        "gigabytes": 1000,
        "snapshot_gigabytes": 999,
        "shares": 50,
        "snapshots": 49,
        "share_networks": 9
    }
}
```

17.18.3. Delete quotas

Method	URI	Description
DELETE	/v2/{tenant_id}/os-quota-sets/{tenant_id}{?user_id}	Deletes quotas for a tenant. The quota will revert back to default.

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

17.18.3.1. Request

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

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{tenant_id}	UUID	The ID 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 admin tenant.

This operation does not accept a request body.

17.18.4. Show default quotas

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

Normal response codes: 200

17.18.4.1. Request

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

Name	Type	Description
X-Openstack-Manila-Api-Version	String <i>(Optional)</i>	The HTTP header to specify a valid Shared File Systems API micro-version. For example, "X-Openstack-Manila-Api-Version: 2.6". 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 tenant ID in a multi-tenancy cloud.
{tenant_id}	UUID	The ID 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 admin tenant.

This operation does not accept a request body.

17.18.4.2. Response

Example 17.82. Show default quotas: JSON response

```
{
    "quota_set": {
        "gigabytes": 1000,
        "shares": 50,
        "snapshot_gigabytes": 1000,
        "snapshots": 50,
        "id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "share_networks": 10
    }
}
```

18. 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 the [Shared File Systems API v2](#). Subsequent API v2 micro-versions, such as v2.1, differ from API v1.

Starting with API v2.0, the API uses Compute-style micro-versions.

19. Telemetry API v2 (CURRENT)

Manages alarms, meters, samples, resources, and capabilities through a set of services:

- [Aodh](#). An alarming 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 constrained by restrictions imposed by the storage driver.

19.1. Alarms

Lists, creates, gets details for, updates, and deletes alarms.

Method	URI	Description
GET	/v2/alarms{?q}	Lists alarms, based on a query.

Method	URI	Description
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.

19.1.1. List alarms

Method	URI	Description
GET	/v2/alarms{?q}	Lists alarms, based on a query.

Normal response codes: 200

19.1.1.1. Request

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

Name	Type	Description
q	List <i>(Optional)</i>	Filter rules for the changes to be described.

This operation does not accept a request body.

19.1.1.2. Response

Example 19.1. 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 19.2. 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.

19.1.2. Create alarm

Method	URI	Description
POST	/v2/alarms{?data}	Creates an alarm.

Normal response codes: 200

19.1.2.1. Request

This table shows the query parameters for the create alarm request:

Name	Type	Description
data	Alarm <i>(Optional)</i>	An alarm within the request body.

This operation does not accept a request body.

19.1.2.2. Response

Example 19.3. 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 19.4. 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" />
```

```
<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.

19.1.3. Show alarm details

Method	URI	Description
GET	/v2/alarms/{alarm_id}	Shows details for an alarm, by alarm ID.

Normal response codes: 200

19.1.3.1. Request

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

Name	Type	Description
{alarm_id}	String	The UUID of the alarm.

This operation does not accept a request body.

19.1.3.2. Response

Example 19.5. 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 19.6. 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.

19.1.4. Update alarm

Method	URI	Description
PUT	/v2/alarms/{alarm_id}{?data}	Updates an alarm.

Normal response codes: 200

19.1.4.1. Request

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

Name	Type	Description
{alarm_id}	String	The UUID of the alarm.

This table shows the query parameters for the update alarm request:

Name	Type	Description
data	Alarm <i>(Optional)</i>	An alarm within the request body.

This operation does not accept a request body.

19.1.4.2. Response

Example 19.7. 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": "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 19.8. 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.

19.1.5. Delete alarm

Method	URI	Description
DELETE	/v2/alarms/{alarm_id}	Deletes an alarm, by alarm ID.

Normal response codes: 204

19.1.5.1. Request

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

Name	Type	Description
{alarm_id}	String	The UUID of the alarm.

This operation does not accept a request body.

19.1.6. Update alarm state

Method	URI	Description
PUT	/v2/alarms/{alarm_id}/state{?state}	Sets the state of an alarm.

Normal response codes: 200

19.1.6.1. Request

This table shows the URI parameters for the update alarm state request:

Name	Type	Description
{alarm_id}	String	The UUID of the alarm.

This table shows the query parameters for the update alarm state request:

Name	Type	Description
state	String <i>(Required)</i>	The alarm state. A valid value is ok, alarm, or insufficient data.

This operation does not accept a request body.

19.1.7. 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

19.1.7.1. Request

This table shows the URI parameters for the show alarm state request:

Name	Type	Description
{alarm_id}	String	The UUID of the alarm.

19.1.8. 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

19.1.8.1. Request

This table shows the URI parameters for the show alarm history request:

Name	Type	Description
{alarm_id}	String	The UUID of the alarm.

This table shows the query parameters for the show alarm history request:

Name	Type	Description
q	List <i>(Optional)</i>	Filter rules for the changes to be described.

This operation does not accept a request body.

19.2. 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": 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,
        "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.

19.2.1. List meters

Method	URI	Description
GET	/v2/meters{?q,limit}	Lists meters, based on the data recorded so far.

Normal response codes: 200

19.2.1.1. Request

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

Name	Type	Description
q	List <i>(Optional)</i>	Filter rules for the changes to be described.
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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.

19.2.1.2. Response

Example 19.9. 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 19.10. 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.

19.2.2. Add samples to meter

Method	URI	Description
POST	/v2/meters/{meter_name}{?direct,samples}	Adds samples to a meter, by meter name.

Normal response codes: 200

19.2.2.1. 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 POSTed directly to storage. Set ?direct=True to POST the samples directly to storage.
samples	List (Optional)	A list of samples.

Example 19.11. 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 19.12. 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.

19.2.2.2. Response

Example 19.13. 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 19.14. 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>
  </metadata>
```

```
</item>
<item>
  <key>name1</key>
  <value>value1</value>
</item>
</metadata>
</value>
```

This operation does not return a response body.

19.2.3. 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

19.2.3.1. 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 <i>(Optional)</i>	Filter rules for the changes to be described.
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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.

19.2.3.2. Response

Example 19.15. 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 19.16. List samples for 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.

19.2.4. 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

19.2.4.1. 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 <i>(Optional)</i>	Filter rules for the changes to be described.
groupby	List <i>(Optional)</i>	Fields for group by aggregation.
period	Int <i>(Optional)</i>	The period, in seconds, for which you want statistics.
aggregate	List <i>(Optional)</i>	A list of selectable aggregation functions to be applied. For example: GET /v2/meters/METER_NAME/statistics?aggregate.func=cardinality&aggregate.param=resource_id&aggregate.func=cardinality&aggregate.param=project_id
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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.

19.2.4.2. Response

Example 19.17. Show meter statistics: JSON response

```
[  
  {  
    "avg": 4.5,  
    "count": 10,  
    "duration": 300.0,  
    "duration_end": "2013-01-04T16:47:00",  
    "duration_start": "2013-01-04T16:42:00",  
    "max": 9.0,  
    "min": 1.0,
```

```
        "period": 7200,
        "period_end": "2013-01-04T18:00:00",
        "period_start": "2013-01-04T16:00:00",
        "sum": 45.0,
        "unit": "GiB"
    }
]
```

Example 19.18. 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.

19.3. 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.

19.3.1. 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

19.3.1.1. Request

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

Name	Type	Description
q	List <i>(Optional)</i>	Filter rules for the changes to be described.
limit	Int <i>(Optional)</i>	Requests a page size of returned items from the query. 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.

19.3.1.2. Response

Example 19.19. 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 19.20. 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.

19.3.2. Show sample details

Method	URI	Description
GET	/v2/samples/{sample_id}	Shows details for a sample, by sample ID.

Normal response codes: 200

19.3.2.1. Request

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

Name	Type	Description
{sample_id}	String	The UUID of the sample.

This operation does not accept a request body.

19.3.2.2. Response

Example 19.21. Show sample details: JSON response

```
{
  "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 19.22. Show sample details: XML response

```
<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 return a response body.

19.4. 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.

19.4.1. List resources

Method	URI	Description
GET	/v2/resources{?q,meter_links}	Lists definitions for all resources.

Normal response codes: 200

19.4.1.1. Request

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

Name	Type	Description
q	List <i>(Optional)</i>	Filter rules for the changes to be described.
meter_links	Int <i>(Optional)</i>	Set ?meter_links=1 to return a self link and related meter links.

This operation does not accept a request body.

19.4.1.2. Response

Example 19.23. 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
	(Required)	
metadata	Dict (Required)	A set of one or more arbitrary metadata key and value pairs that are associated with the resource.
project_id	String (Required)	The ID of the owning project or tenant.
resource_id	String (Required)	The unique identifier for the resource.
source	String (Required)	The name of the source from which the resource came.
user_id	String (Required)	The ID of the user who either created or last updated the resource.

Example 19.24. 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&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.

19.4.2. Show resource details

Method	URI	Description
GET	/v2/resources/{resource_id}	Shows details for a resource, by resource ID.

Normal response codes: 200

19.4.2.1. Request

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

Name	Type	Description
{resource_id}	String	The UUID of the resource.

This operation does not accept a request body.

19.4.2.2. Response

Example 19.25. 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 19.26. 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.

19.5. 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 constrained by restrictions imposed by the storage driver.

19.5.1. List capabilities

Method	URI	Description
GET	/v2/capabilities	A representation of the API and storage capabilities. Usually constrained by restrictions imposed by the storage driver.

Normal response codes: 200

19.5.1.1. Response

Example 19.27. 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	Array	An array of API capabilities for the configured storage driver.

Name	Type	Description
	(Required)	
alarms:history:query:complex	Boolean (Required)	Indicates whether the complex query capability for alarm history is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
alarms:history:query:simple	Boolean (Required)	Indicates whether the simple query capability for alarm history is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
alarms:query:complex	Boolean (Required)	Indicates whether the complex query capability for alarm definitions is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
alarms:query:simple	Boolean (Required)	Indicates whether the simple query capability for alarm definitions is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
events:query:simple	Boolean (Required)	Indicates whether the simple query capability for events is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
meters:query:complex	Boolean (Required)	Indicates whether the complex query capability for meters is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
meters:query:metadata	Boolean (Required)	Indicates whether the simple query capability for the metadata of meters is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
meters:query:simple	Boolean (Required)	Indicates whether the simple query capability for meters is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
resources:query:complex	Boolean (Required)	Indicates whether the complex query capability for resources is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
resources:query:metadata	Boolean (Required)	Indicates whether the simple query capability for the metadata of resources is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
resources:query:simple	Boolean (Required)	Indicates whether the simple query capability for resources is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
samples:query:complex	Boolean (Required)	Indicates whether the complex query capability for samples is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
samples:query:metadata	Boolean (Required)	Indicates whether the simple query capability for the metadata of samples is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
samples:query:simple	Boolean (Required)	Indicates whether the simple query capability for samples is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
statistics:aggregation:select:avg	Boolean (Required)	Indicates whether the <code>avg</code> capability is available for the configured database back end. A value of <code>true</code> indicates that the capability is available. Use the <code>avg</code> capability to get average values for samples.
statistics:aggregation:select:cardinality	Boolean (Required)	Indicates whether the <code>cardinality</code> capability is available for the configured database back end. A value of <code>true</code> indicates that the capability is available. Use the <code>cardinality</code> capability to get cardinality for samples.
statistics:aggregation:select:count	Boolean (Required)	Indicates whether the <code>count</code> capability is available for the configured database back end. A value of <code>true</code> indicates that the capability is available. Use the <code>count</code> capability to calculate the number of samples for a query.
statistics:aggregation:select:max	Boolean (Required)	Indicates whether the <code>max</code> capability is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.

Name	Type	Description
		available. Use the <code>max</code> capability to calculate the maximum value for a query.
<code>statistics:aggregation:sel_min</code>	Boolean (Required)	Indicates whether the <code>min</code> capability is available for the configured database back end. A value of <code>true</code> indicates that the capability is available. Use the <code>min</code> capability to calculate the minimum value for a query.
<code>statistics:aggregation:sel_quartile</code>	Boolean (Required)	Indicates whether the <code>quartile</code> capability is available for the configured database back end. A value of <code>true</code> indicates that the capability is available. Use the <code>quartile</code> capability to calculate the quartile of sample volumes for a query.
<code>statistics:aggregation:sel_stddev</code>	Boolean (Required)	Indicates whether the <code>stddev</code> capability is available for the configured database back end. A value of <code>true</code> indicates that the capability is available. Use the <code>stddev</code> capability to calculate the standard deviation of sample volumes for a query.
<code>statistics:aggregation:sel_sum</code>	Boolean (Required)	Indicates whether the <code>sum</code> capability is available for the configured database back end. A value of <code>true</code> indicates that the capability is available. Use the <code>sum</code> capability to calculate the sum of sample volumes for a query.
<code>statistics:aggregation:standard_set</code>	Boolean (Required)	Indicates whether the standard set of aggregation capability is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
<code>statistics:groupby</code>	Boolean (Required)	Indicates whether the <code>groupby</code> capability is available for calculating statistics for the configured database back end. A value of <code>true</code> indicates that the capability is available.
<code>statistics:query:complex</code>	Boolean (Required)	Indicates whether the complex query capability for statistics is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
<code>statistics:query:metadata</code>	Boolean (Required)	Indicates whether the simple query capability for the metadata of samples used for calculating statistics is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
<code>statistics:query:simple</code>	Boolean (Required)	Indicates whether the simple query capability for statistics is available for the configured database back end. A value of <code>true</code> indicates that the capability is available.
<code>alarm_storage</code>	Array (Required)	Defines the capabilities for the storage that stores persisting alarm definitions. A value of <code>true</code> indicates that the capability is available.
<code>storage:production_ready</code>	Boolean (Required)	Indicates whether the configured database back end is ready to use in production environment. A value of <code>true</code> indicates that the database back end is ready to use in a production environment.
<code>event_storage</code>	Array (Required)	Defines the capabilities for the storage that stores persisting events. A value of <code>true</code> indicates that the capability is available.
<code>storage</code>	Array (Required)	Defines the capabilities for the storage that stores persisting samples. A value of <code>true</code> indicates that the capability is available.
<code>storage:production_ready</code>	Boolean (Required)	Indicates whether the configured database back end is ready to use in production environment. A value of <code>true</code> indicates that the database back end is ready to use in a production environment.

Example 19.28. 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>
<item>
    <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.