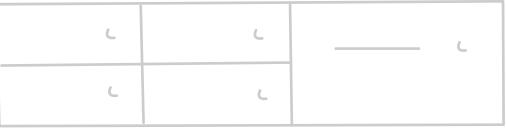
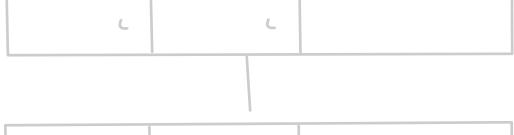
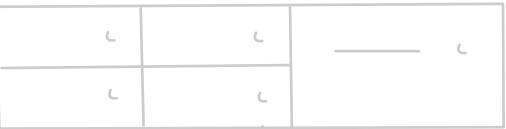


# OpenStack

## API Complete Reference

(August 5, 2014)



## OpenStack API Complete Reference

Copyright © 2010-2014 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 .....	2
1.1.1. List API versions .....	3
1.1.2. Show API version details .....	4
1.2. API extensions .....	4
1.2.1. List API extensions .....	6
1.3. Volumes .....	13
1.3.1. Create volume .....	14
1.3.2. List volumes .....	17
1.3.3. List volumes (detailed) .....	19
1.3.4. Show volume information .....	22
1.3.5. Update volume .....	24
1.3.6. Delete volume .....	26
1.4. Volume types .....	26
1.4.1. List volume types .....	27
1.4.2. Create volume type .....	28
1.4.3. Show volume type information .....	30
1.4.4. Delete volume type .....	31
1.5. Snapshots .....	31
1.5.1. Create snapshot .....	32
1.5.2. List snapshots .....	34
1.5.3. List snapshots (detailed) .....	36
1.5.4. Show snapshot information .....	38
1.5.5. Update snapshot .....	39
1.5.6. Delete snapshot .....	41
1.5.7. Show snapshot metadata .....	42
1.5.8. Update snapshot metadata .....	44
1.6. Quality of service (QoS) specifications (qos-specs) .....	45
1.6.1. Create QoS specification .....	46
1.6.2. List QoS specs .....	48
1.6.3. Show QoS specification details .....	50
1.6.4. Delete QoS specification .....	52
1.6.5. Associate QoS specification with volume type .....	53
1.6.6. Disassociate QoS specification from volume type .....	54
1.6.7. Disassociate QoS specification from all associations .....	55
1.6.8. Get all associations for QoS specification .....	56
1.7. Quota sets extension (os-quota-sets) .....	56
1.7.1. Show quotas .....	58
1.7.2. Update quotas .....	60
1.7.3. Delete quotas .....	63
1.7.4. Get default quotas .....	64
1.7.5. Show quotas for user .....	66
1.7.6. Update quotas for user .....	68
1.7.7. Delete quotas for user .....	71
1.7.8. Show quota details for user .....	72
1.8. Limits extension (limits) .....	73
1.8.1. Show absolute limits .....	74
2. Block Storage API v1 (CURRENT) .....	75

2.1. API versions .....	75
2.1.1. List API versions .....	76
2.1.2. Show API v1 details .....	77
2.2. Volumes .....	77
2.2.1. Create volume .....	79
2.2.2. List volumes .....	81
2.2.3. List volumes (detailed) .....	83
2.2.4. Show volume information .....	85
2.2.5. Delete volume .....	87
2.3. Volume types .....	87
2.3.1. List volume types .....	88
2.3.2. Create volume type .....	89
2.3.3. Show volume type information .....	91
2.3.4. Delete volume type .....	92
2.4. Snapshots .....	92
2.4.1. Create snapshot .....	93
2.4.2. List snapshots .....	95
2.4.3. List snapshots (detailed) .....	97
2.4.4. Show snapshot information .....	99
2.4.5. Delete snapshot .....	100
2.4.6. Show snapshot metadata .....	101
2.4.7. Update snapshot metadata .....	103
3. Compute API v2 (CURRENT) .....	104
3.1. API versions .....	105
3.1.1. List API versions .....	107
3.1.2. Show API version details .....	109
3.2. Limits .....	110
3.2.1. List limits .....	111
3.3. Extensions .....	113
3.3.1. List extensions .....	114
3.3.2. Get extension .....	137
3.4. Servers .....	138
3.4.1. List servers .....	139
3.4.2. Create server .....	141
3.4.3. List details for servers .....	145
3.4.4. Get server details .....	149
3.4.5. Update server .....	152
3.4.6. Delete server .....	159
3.5. Server metadata .....	159
3.5.1. Show server metadata .....	160
3.5.2. Create or replace server metadata .....	161
3.5.3. Update server metadata items .....	163
3.5.4. Show server metadata item details .....	165
3.5.5. Create or update server metadata item .....	166
3.5.6. Delete server metadata item .....	168
3.6. Server addresses .....	168
3.6.1. List addresses .....	169
3.6.2. List addresses by network .....	171
3.7. Server actions .....	172
3.7.1. Change password .....	173
3.7.2. Reboot server .....	174

3.7.3. Rebuild server .....	175
3.7.4. Resize server .....	178
3.7.5. Confirm resized server .....	179
3.7.6. Revert resized server .....	180
3.7.7. Create image .....	181
3.8. Flavors .....	182
3.8.1. List flavors .....	183
3.8.2. List details for flavors .....	187
3.8.3. Get flavor details .....	190
3.9. Images .....	191
3.9.1. List images .....	192
3.9.2. List images details .....	197
3.9.3. Get image details .....	204
3.9.4. Delete image .....	206
3.10. Image metadata .....	206
3.10.1. Show image metadata .....	207
3.10.2. Create or replace image metadata .....	209
3.10.3. Update image metadata items .....	211
3.10.4. Show image metadata item details .....	213
3.10.5. Create or update image metadata item .....	214
3.10.6. Delete image metadata item .....	216
4. Compute API v2 extensions (CURRENT) .....	217
4.1. Server admin actions (action) .....	225
4.1.1. Pause server .....	227
4.1.2. Unpause server .....	228
4.1.3. Suspend server .....	229
4.1.4. Resume server .....	230
4.1.5. Migrate server .....	231
4.1.6. Reset networking on server .....	232
4.1.7. Inject network information .....	233
4.1.8. Lock server .....	234
4.1.9. Unlock server .....	235
4.1.10. Create server backup .....	236
4.1.11. Live-migrate server .....	237
4.1.12. Reset server state .....	238
4.1.13. Evacuate server .....	239
4.1.14. Add security group .....	241
4.1.15. Remove security group .....	242
4.1.16. Add floating IP address .....	243
4.2. Server diagnostics (diagnostics) .....	243
4.2.1. Get server diagnostics .....	244
4.3. Flavor access (flavors) .....	244
4.3.1. List flavors with access type .....	246
4.3.2. Create private flavor .....	250
4.3.3. Show flavor access type .....	253
4.3.4. List tenants with access to private flavor .....	255
4.3.5. Add access to private flavor .....	257
4.3.6. Delete access from private flavor .....	259
4.4. Flavors with FlavorDisabled attribute (flavors) .....	260
4.4.1. Get flavor disabled status details .....	261
4.4.2. List flavors with flavor disabled status .....	263

4.5. Flavor extra-specs (os-extra-specs) .....	265
4.5.1. List flavor extra specs .....	267
4.5.2. Create flavor extra specs .....	268
4.5.3. Get flavor extra spec details .....	270
4.5.4. Delete flavor extra specs .....	271
4.6. Flavors with rxtx_factor extended attribute (flavors) .....	271
4.6.1. Create flavor with rxtx_factor .....	272
4.6.2. Get flavor with rxtx_factor .....	274
4.6.3. Get flavor Details with rxtx_factor .....	276
4.7. Flavors with extended attributes (flavors) .....	278
4.7.1. Create flavor with extra data .....	280
4.7.2. Get flavor extra data details .....	282
4.7.3. List flavors with extra data .....	284
4.8. Flavors create or delete (flavors) .....	286
4.8.1. Create flavor .....	288
4.8.2. Delete flavor .....	290
4.9. Images with size attribute (images) .....	290
4.9.1. List details for images .....	291
4.9.2. Get image details .....	298
4.10. Limits with project usage (limits) .....	299
4.10.1. Get limits .....	300
4.11. Limits with project usage for administrators (limits) .....	302
4.11.1. Get customer limits .....	304
4.12. Guest agents (os-agents) .....	306
4.12.1. List agents .....	308
4.12.2. Create agent .....	309
4.12.3. Delete agent .....	311
4.12.4. Update agent .....	312
4.13. Host aggregates (os-aggregates) .....	313
4.13.1. List aggregates .....	314
4.13.2. Create aggregate .....	316
4.13.3. Delete aggregate .....	318
4.13.4. Get aggregate details .....	319
4.13.5. Update aggregate .....	321
4.13.6. Set aggregate metadata .....	323
4.13.7. Add host to aggregate .....	325
4.13.8. Remove host from aggregate .....	327
4.14. Attach interfaces (os-interface) .....	328
4.14.1. Create interface .....	329
4.14.2. List interfaces .....	331
4.14.3. Show attached interface information .....	333
4.14.4. Detach interface .....	335
4.15. Root certificates (os-certificates) .....	335
4.15.1. Create root certificate .....	336
4.15.2. Show root certificate details .....	339
4.16. Cloupipe (os-cloupipe) .....	340
4.16.1. List cloupipes .....	341
4.16.2. Create cloupipe .....	342
4.16.3. Update cloupipe .....	343
4.17. Server console output (os-console-output) .....	343
4.17.1. Get console output for an instance .....	344

4.18. Server console (os-consoles) .....	345
4.18.1. Get console .....	346
4.19. Coverage reports (os-coverage) .....	347
4.19.1. Get coverage report .....	348
4.19.2. Start coverage report .....	350
4.19.3. Start combined coverage report .....	351
4.19.4. Stop coverage report .....	352
4.20. Server deferred delete (os-deferred-delete) .....	352
4.20.1. Force delete server .....	354
4.20.2. Restore server .....	355
4.21. Fixed IPs (os-fixed-ips) .....	355
4.21.1. Show fixed IP information .....	356
4.21.2. Reserve or release a fixed IP .....	357
4.22. Floating IP DNS records (os-floating-ip-dns) .....	357
4.22.1. List DNS domains .....	358
4.22.2. Create or update DNS domain .....	359
4.22.3. Delete DNS domain .....	361
4.22.4. Create or update DNS entry .....	362
4.22.5. Find unique DNS entry .....	364
4.22.6. Delete DNS entry .....	365
4.22.7. List DNS entries .....	366
4.23. Floating IP pools (os-floating-ip-pools) .....	366
4.23.1. List floating IP pools .....	367
4.24. Floating IPs (os-floating-ips) .....	367
4.24.1. List floating IPs .....	369
4.24.2. Allocate floating IP .....	370
4.24.3. Show floating IP information .....	372
4.24.4. Deallocate floating IP .....	373
4.24.5. Add floating IP .....	374
4.24.6. Remove floating IP .....	375
4.25. Floating IPs bulk (os-floating-ips-bulk) .....	375
4.25.1. List floating IPs .....	377
4.25.2. Create floating IPs .....	379
4.25.3. Bulk-delete floating IPs .....	381
4.25.4. List floating IPs by host .....	382
4.26. Hosts (os-hosts) .....	382
4.26.1. List hosts .....	384
4.26.2. Show host information .....	386
4.26.3. Update host .....	388
4.26.4. Start host .....	390
4.26.5. Shut down host .....	391
4.26.6. Reboot host .....	392
4.27. Hypervisors (os-hypervisors) .....	392
4.27.1. List hypervisors .....	394
4.27.2. Show hypervisor information .....	395
4.27.3. Show statistics for hypervisors .....	397
4.27.4. Show hypervisor up time .....	398
4.27.5. List instances for hypervisors .....	399
4.28. Server actions (os-instance-actions) .....	399
4.28.1. List server actions .....	401
4.28.2. Get action details .....	403

4.29. Keypairs (os-keypairs) .....	404
4.29.1. List keypairs .....	405
4.29.2. Create or import keypair .....	406
4.29.3. Delete keypair .....	408
4.29.4. Show keypair information .....	409
4.30. Migrations (os-migrations) .....	409
4.30.1. Get migrations .....	410
4.31. Networks (os-networks) .....	411
4.31.1. Create network .....	413
4.31.2. List networks .....	416
4.31.3. Add network .....	419
4.31.4. Show network information .....	420
4.31.5. Delete network .....	422
4.31.6. Associate host .....	423
4.31.7. Disassociate host .....	424
4.31.8. Disassociate network .....	425
4.31.9. Disassociate project .....	426
4.32. Quota sets (os-quota-sets) .....	426
4.32.1. Show quotas .....	427
4.32.2. Update quotas .....	429
4.32.3. Delete quotas .....	432
4.32.4. Get default quotas .....	433
4.32.5. Show quotas for user .....	435
4.32.6. Update quotas for user .....	437
4.32.7. Delete quotas for user .....	440
4.32.8. Show quota details for user .....	441
4.33. Rules for default security group (os-security-group-default-rules) .....	442
4.33.1. List default security group rules .....	443
4.33.2. Create default security group rule .....	445
4.33.3. Show default security group rule information .....	447
4.33.4. Delete default security group rule .....	449
4.34. Security groups (os-security-groups) .....	449
4.34.1. List security groups .....	450
4.34.2. Create security group .....	451
4.34.3. List security groups by server .....	453
4.34.4. Show security group information .....	454
4.34.5. Delete security group .....	455
4.35. Server groups (os-server-groups) .....	455
4.35.1. List server groups .....	456
4.35.2. Create server group .....	458
4.35.3. Show server group details .....	460
4.35.4. Delete server group .....	462
4.36. Server password (os-server-password) .....	462
4.36.1. Get server password .....	463
4.36.2. Clear server password .....	464
4.37. Server shelve (servers) .....	465
4.37.1. Shelve server .....	467
4.37.2. Remove a shelved instance .....	468
4.37.3. Restore shelved server .....	469
4.38. Server start and stop (servers) .....	469
4.38.1. Start server .....	470

4.38.2. Stop server .....	471
4.39. Manage services (os-services) .....	471
4.39.1. List services .....	472
4.39.2. Enable scheduling for a service .....	474
4.39.3. Disable scheduling for a service .....	475
4.39.4. Log disabled service information .....	476
4.39.5. List disabled services .....	478
4.40. Usage reports (os-simple-tenant-usage) .....	479
4.40.1. List usage information for all tenants .....	480
4.40.2. Get tenant usage information .....	481
4.41. Virtual interfaces (os-virtual-interfaces) .....	482
4.41.1. List virtual interfaces .....	483
4.41.2. Show virtual interface and attached network .....	484
4.42. Volume extension (os-volumes, os-snapshots) .....	484
4.42.1. List volumes .....	486
4.42.2. List details for volumes .....	488
4.42.3. Create volume .....	490
4.42.4. Show volume information .....	492
4.42.5. Delete volume .....	494
4.42.6. List volume types .....	495
4.42.7. Show volume type .....	496
4.42.8. Create snapshot .....	497
4.42.9. List snapshots .....	499
4.42.10. List details for snapshots .....	501
4.42.11. Show snapshot .....	503
4.42.12. Delete snapshot .....	504
4.43. Volume attachments (os-volume_attachments) .....	504
4.43.1. Attach volume .....	505
4.43.2. List volume attachments .....	507
4.43.3. Show volume attachment details .....	509
4.43.4. Delete volume attachment .....	510
4.44. Servers with block device mapping format (servers) .....	510
4.44.1. List servers .....	511
4.44.2. Create server .....	513
4.45. Server OS-EXT-IPS-MAC:mac_addr extended attribute (servers) .....	517
4.45.1. Create server with OS-EXT-IPS-MAC:mac_addr extended attribute.....	518
4.45.2. Show server information .....	521
4.45.3. Get server details .....	524
4.46. Configuration drive (servers) .....	526
4.46.1. Create server with configuration drive .....	527
4.46.2. Get server information with configuration drive .....	530
4.46.3. Get server details with configuration drive .....	534
4.47. Servers with extended availability zones (servers) .....	537
4.47.1. Show server .....	538
4.47.2. List details for servers .....	540
4.48. Servers and images with disk config (servers, images) .....	542
4.48.1. Create server .....	543
4.48.2. Show server information .....	547
4.48.3. Update server .....	550
4.48.4. Resize server .....	553
4.48.5. Rebuild server .....	554

4.48.6. List servers .....	557
4.48.7. Get image information .....	560
4.48.8. List images .....	562
4.49. Server IP type (servers) .....	568
4.49.1. Show IP type .....	569
4.49.2. List servers with IP type .....	571
4.50. Server extended attributes (servers) .....	573
4.50.1. List servers with extended server attributes .....	574
4.50.2. Show extended server attributes .....	577
4.51. Server extended status (servers) .....	579
4.51.1. Show server extended status .....	580
4.51.2. List extended status for servers .....	583
4.52. Servers multiple create (servers) .....	585
4.52.1. Create multiple servers .....	586
4.53. Server rescue and unrescue (servers) .....	591
4.53.1. Rescue server .....	592
4.53.2. Unrescue server .....	594
4.54. Servers with scheduler hints (servers) .....	594
4.54.1. Create server with scheduler hints .....	595
5. Compute API v3 (EXPERIMENTAL) .....	598
5.1. API versions .....	601
5.1.1. List API versions .....	602
5.1.2. Show API v3 details .....	604
5.2. Server admin actions (servers) .....	605
5.2.1. Back up server .....	607
5.2.2. Inject network information .....	608
5.2.3. Live migrate to new host .....	609
5.2.4. Lock server .....	610
5.2.5. Migrate server .....	611
5.2.6. Pause server .....	612
5.2.7. Reset server networking .....	613
5.2.8. Resume server .....	614
5.2.9. Suspend server .....	615
5.2.10. Unlock server .....	616
5.2.11. Unpause server .....	617
5.3. Guest agents (os-agents) .....	617
5.3.1. Create agent build .....	618
5.3.2. List agent builds .....	620
5.3.3. Update agent build .....	621
5.4. Host aggregates (os-aggregates) .....	622
5.4.1. Create aggregate .....	623
5.4.2. List aggregates .....	624
5.4.3. Show aggregate details .....	626
5.4.4. Update aggregate .....	628
5.4.5. Add host .....	630
5.4.6. Create or update aggregate metadata .....	632
5.4.7. Remove host .....	634
5.5. Cells (os-cells) .....	635
5.5.1. List cells .....	636
5.5.2. Lists cells with details .....	638
5.5.3. Shows cell data .....	639

5.5.4. Shows cell capacities .....	640
5.6. Root certificates (os-certificates) .....	641
5.6.1. Create certificate .....	642
5.6.2. Show certificate details .....	645
5.7. Configuration drive (os-config-drive) .....	646
5.7.1. Show server details .....	647
5.7.2. List server details for a user .....	649
5.8. Server deferred delete (os-deferred-delete) .....	650
5.8.1. Force delete instance .....	652
5.8.2. Restore deleted instance .....	653
5.9. Evacuate (os-evacuate) .....	653
5.9.1. Evacuate server .....	654
5.10. Servers with extended availability zones (os-extended-availability-zone) .....	654
5.10.1. Show server details .....	656
5.10.2. List server details for user .....	658
5.11. Server extended attributes (os-extended-server-attributes) .....	659
5.11.1. Shows server details .....	661
5.11.2. Lists server details for user .....	663
5.12. Server extended status (os-extended-status) .....	665
5.12.1. Show server details .....	666
5.12.2. List servers with details for user .....	668
5.13. Flavor access (os-flavor-access) .....	670
5.13.1. Add access attribute to flavor create .....	671
5.13.2. Add access attribute to flavor detail .....	673
5.13.3. Add access attribute to flavor show .....	676
5.13.4. Add flavor access .....	677
5.13.5. Remove flavor access .....	678
5.13.6. Return access list .....	679
5.14. Flavor extra-specs (flavor-extra-specs) .....	679
5.14.1. Create or update flavor extra specs .....	681
5.14.2. Show flavor extra specs .....	682
5.14.3. Show flavor extra specs .....	683
5.14.4. Update flavor extra spec .....	684
5.15. Flavors manage (flavor-manage) .....	684
5.15.1. Create flavor .....	685
5.15.2. Delete flavor .....	687
5.16. Flavors with rxtx_factor extended attribute (os-flavor-rxtx) .....	687
5.16.1. Create flavor .....	688
5.16.2. List flavors with details .....	690
5.16.3. Show flavor details .....	693
5.17. Flavors with extended attributes (flavors) .....	693
5.17.1. List flavors .....	695
5.17.2. Show flavor details .....	697
5.17.3. List detailed flavors .....	698
5.18. Hosts (os-hosts) .....	700
5.18.1. List hosts .....	701
5.18.2. Enables host .....	703
5.18.3. Show host details .....	704
5.18.4. Reboot host .....	706
5.18.5. Shut down host .....	707
5.18.6. Start host .....	708

5.19. Hypervisors (os-hypervisors) .....	708
5.19.1. List hypervisors .....	709
5.19.2. Show hypervisor statistics .....	710
5.19.3. Search hypervisors .....	711
5.19.4. Show hypervisor details .....	712
5.19.5. Show hypervisor uptime .....	713
5.19.6. List servers for a hypervisor .....	714
5.20. Server actions (servers) .....	714
5.20.1. List actions for an instance .....	716
5.20.2. Shows instance action .....	717
5.21. Instance usage audit log (os-instance-usage-audit-log) .....	718
5.21.1. List usage audits .....	719
5.21.2. List instance usage .....	720
5.22. Limits (limits) .....	721
5.22.1. Show limits .....	722
5.23. Migrations (os-migrations) .....	723
5.23.1. List migrations .....	724
5.24. Multinic (os-multinic) .....	725
5.24.1. Remove IP from instance .....	726
5.24.2. Add IP to instance .....	727
5.25. Quota class (os-quota-class-sets) .....	727
5.25.1. Show quota .....	728
5.25.2. Update quota .....	729
5.26. Quota sets (os-quota-sets) .....	730
5.26.1. Delete quota for tenant .....	731
5.26.2. Show quotas .....	732
5.26.3. Force-update quota .....	733
5.26.4. Update quota .....	735
5.26.5. Show default quotas .....	736
5.26.6. Update quota for user .....	737
5.26.7. Delete quota for user .....	739
5.27. Server remote console (os-remote-consoles) .....	739
5.27.1. Get output for spice .....	740
5.27.2. Get console output .....	741
5.28. Server usage (os-server-usage) .....	741
5.28.1. List server details for user .....	743
5.28.2. Show server details .....	745
5.29. Usage reports (os-simple-tenant-usage) .....	746
5.29.1. List tenant usage for all tenants .....	748
5.29.2. Show usage details for tenant .....	749
6. Databases Service API v1.0 (CURRENT) .....	751
6.1. API versions .....	751
6.1.1. List versions .....	752
6.1.2. Show version details .....	753
6.2. Database instances (instances) .....	753
6.2.1. Create database instance .....	755
6.2.2. List database instances .....	758
6.2.3. Show database instance details .....	760
6.2.4. Delete database instance .....	762
6.2.5. Enable root user .....	763
6.2.6. Show root-enabled status for database instance .....	765

6.3. Database instance actions (action) .....	766
6.3.1. Restart instance .....	767
6.3.2. Resize instance .....	768
6.3.3. Resize instance volume .....	769
6.4. Databases (databases) .....	770
6.4.1. Create database .....	771
6.4.2. List instance databases .....	773
6.4.3. Delete database .....	775
6.5. Users (users) .....	775
6.5.1. Create user .....	777
6.5.2. List database instance users .....	780
6.5.3. Delete user .....	782
6.6. Flavors (flavors) .....	783
6.6.1. List flavors .....	784
6.6.2. Show flavor details .....	786
7. Identity API v3 (STABLE) .....	788
7.1. API versions .....	790
7.1.1. List versions .....	791
7.1.2. Show API version details .....	794
7.2. Tokens .....	795
7.2.1. Authenticate .....	796
7.2.2. Validate token .....	803
7.2.3. Check token .....	805
7.2.4. Revoke token .....	806
7.3. Service catalog .....	806
7.3.1. Add service .....	807
7.3.2. List services .....	808
7.3.3. Show service details .....	810
7.3.4. Update service .....	811
7.3.5. Delete service .....	812
7.4. Endpoints .....	812
7.4.1. Add endpoint .....	813
7.4.2. List endpoints .....	814
7.4.3. Update endpoint .....	816
7.4.4. Delete endpoint .....	818
7.5. Domains .....	818
7.5.1. Add domain .....	820
7.5.2. List domains .....	821
7.5.3. Show domain details .....	823
7.5.4. Update domain .....	824
7.5.5. Delete domain .....	825
7.5.6. List roles for domain user .....	826
7.5.7. List roles for domain group .....	827
7.5.8. Grant role to domain group .....	828
7.5.9. Check role for domain group .....	829
7.5.10. Revoke role from domain group .....	830
7.6. Projects .....	830
7.6.1. Add project .....	832
7.6.2. List projects .....	833
7.6.3. Show project details .....	835
7.6.4. Update project .....	836

7.6.5. Delete project .....	837
7.6.6. List roles for project user .....	838
7.6.7. Grant role to project user .....	839
7.6.8. Check role for project user .....	840
7.6.9. Revoke role from project user .....	841
7.6.10. List roles for project group .....	842
7.6.11. Grant role to project group .....	843
7.6.12. Check role for project group .....	844
7.6.13. Revoke Role from Project Group .....	845
7.7. Users .....	845
7.7.1. Add user .....	846
7.7.2. List users .....	847
7.7.3. Show user details .....	849
7.7.4. Update user .....	850
7.7.5. Delete user .....	852
7.7.6. List groups for user .....	853
7.7.7. List projects for user .....	855
7.7.8. List roles for user .....	857
7.8. Groups .....	857
7.8.1. Add group .....	859
7.8.2. List groups .....	860
7.8.3. Show group details .....	862
7.8.4. Update group .....	863
7.8.5. Delete group .....	864
7.8.6. List users in group .....	865
7.8.7. Add user to group .....	867
7.8.8. Remove user from group .....	868
7.8.9. Check user membership in group .....	869
7.9. Credentials .....	869
7.9.1. Add credential .....	870
7.9.2. List credentials .....	871
7.9.3. Show credential details .....	873
7.9.4. Update credential .....	874
7.9.5. Delete credential .....	875
7.10. Roles .....	875
7.10.1. Add role .....	876
7.10.2. List roles .....	877
7.10.3. Grant role to domain user .....	879
7.10.4. Check role for domain user .....	880
7.10.5. Revoke role from domain user .....	881
7.10.6. List role assignments .....	882
7.11. Policies .....	884
7.11.1. Add policy .....	885
7.11.2. List policies .....	886
7.11.3. Show policy details .....	888
7.11.4. Update policy .....	889
7.11.5. Delete policy .....	891
8. Identity API v3 extensions (STABLE) .....	892
8.1. OS-OAUTH1 extension .....	892
8.1.1. Create consumer .....	894
8.1.2. List consumers .....	895

8.1.3. Show consumer .....	896
8.1.4. Update consumer .....	897
8.1.5. Delete consumer .....	898
8.1.6. Create access token .....	899
8.1.7. List authorized access tokens .....	900
8.1.8. Get authorized access token .....	901
8.1.9. Revoke access token .....	902
8.1.10. List roles for an access token .....	903
8.1.11. Get role information for an access token .....	904
8.1.12. Get an Identity Service token .....	905
9. Identity API v2.0 (STABLE) .....	906
9.1. API versions .....	906
9.1.1. List versions .....	907
9.1.2. Show version details .....	909
9.2. Extensions .....	910
9.2.1. List extensions .....	911
9.2.2. Get extension details .....	914
9.3. Tokens .....	915
9.3.1. Authenticate .....	916
9.3.2. List tenants .....	925
10. Identity admin API v2.0 (STABLE) .....	927
10.1. Versions .....	927
10.1.1. Get version details .....	928
10.2. Extensions .....	929
10.2.1. List extensions .....	930
10.2.2. Get extension details .....	933
10.3. Tokens .....	934
10.3.1. Authenticate for Admin API .....	935
10.3.2. Validate token .....	944
10.3.3. Validate token (admin) .....	946
10.4. Users .....	946
10.4.1. Add user .....	947
10.4.2. Update user .....	949
10.4.3. Delete user .....	951
10.4.4. Get user information by name .....	952
10.4.5. Get user information by ID .....	953
10.4.6. List user global roles .....	954
10.5. Tenants .....	954
10.5.1. List tenants .....	956
10.5.2. Get tenant information by name .....	958
10.5.3. Get tenant information by ID .....	960
10.5.4. List roles for user .....	961
11. Identity API v2.0 extensions (STABLE) .....	962
11.1. HP-IDM-serviceld extended parameter .....	964
11.1.1. Validate token .....	965
11.1.2. Validate token for performance .....	967
11.2. OS-KSADM admin extension .....	967
11.2.1. List Users .....	969
11.2.2. Add User .....	971
11.2.3. Update User .....	973
11.2.4. Delete User .....	975

11.2.5. List global roles for user .....	976
11.2.6. Add global role to user .....	977
11.2.7. Delete Global Role from User .....	978
11.2.8. Add Tenant .....	979
11.2.9. Update Tenant .....	981
11.2.10. Delete Tenant .....	983
11.2.11. List Users for a Tenant .....	984
11.2.12. Add Roles to User on Tenant .....	986
11.2.13. Delete Roles from User on Tenant .....	987
11.2.14. Get role information by name .....	988
11.2.15. List Roles .....	989
11.2.16. Add Role .....	990
11.2.17. Get role information .....	992
11.2.18. Delete Role .....	993
11.2.19. List Services .....	994
11.2.20. Add Service .....	996
11.2.21. Get Service by Name .....	998
11.2.22. Get Service .....	999
11.2.23. Delete Service .....	1000
11.3. OS-KSCATALOG admin extension .....	1000
11.3.1. List endpoints .....	1001
11.3.2. Add endpoint .....	1004
11.3.3. Get endpoint .....	1006
11.3.4. Delete endpoint .....	1008
11.3.5. List endpoint templates .....	1009
11.3.6. Add endpoint template .....	1012
11.3.7. Get endpoint template .....	1014
11.3.8. Update endpoint template .....	1016
11.3.9. Delete endpoint template. ....	1018
11.4. OS-KSEC2 admin extension .....	1018
11.4.1. List Credentials .....	1019
11.4.2. Add User Credentials .....	1021
11.4.3. Get User Credentials .....	1023
11.4.4. Update User Credentials .....	1024
11.4.5. Delete User Credentials .....	1026
11.4.6. List Credentials by Type .....	1027
11.5. OS-KSS3 admin extension .....	1027
11.5.1. List credentials .....	1029
11.5.2. Add user credential .....	1031
11.5.3. Get user credentials .....	1033
11.5.4. Update user credentials .....	1034
11.5.5. Delete user credentials .....	1036
11.5.6. List credentials by type .....	1037
11.6. OS-KVALIDATE admin extension .....	1038
11.6.1. Validate token .....	1039
11.6.2. Check token .....	1041
11.6.3. List endpoints for a token .....	1042
11.7. RAX-GRPADMIN admin extension .....	1045
11.7.1. List groups .....	1046
11.7.2. Add group .....	1048
11.7.3. Get group information .....	1050

11.7.4. Update group .....	1051
11.7.5. Delete group .....	1053
11.7.6. List users for group .....	1054
11.7.7. Add user to group .....	1056
11.7.8. Remove user from group .....	1057
11.8. RAX-KSGRP admin extension .....	1057
11.8.1. List groups for a user .....	1058
11.9. RAX-KSKEY admin extension .....	1059
11.9.1. Add user credential .....	1060
11.9.2. List credentials .....	1062
11.9.3. Update user credentials .....	1064
11.9.4. Delete user credentials .....	1066
11.9.5. Get user credentials .....	1067
11.10. RAX-KSQA admin extension .....	1067
11.10.1. Get user secret question and answer .....	1069
11.10.2. Update user secret question and answer .....	1070
12. Image Service API v2 (SUPPORTED) .....	1072
12.1. API versions .....	1072
12.1.1. List API versions .....	1073
12.2. Images .....	1074
12.2.1. Create image .....	1075
12.2.2. List images .....	1076
12.2.3. Get image details .....	1079
12.2.4. Update image .....	1080
12.2.5. Delete image .....	1082
12.3. Image data .....	1082
12.3.1. Upload binary image data .....	1083
12.3.2. Download binary image data .....	1084
12.4. Image tags .....	1084
12.4.1. Add image tag .....	1085
12.4.2. Delete image tag .....	1086
12.5. Members .....	1086
12.5.1. Create image member .....	1087
12.5.2. List image members .....	1088
12.5.3. Show image member details .....	1089
12.5.4. Delete image member .....	1090
12.5.5. Update image member .....	1091
12.6. Image schemas .....	1091
12.6.1. Get images schema .....	1092
12.6.2. Get image schema .....	1097
12.6.3. Get image members schema .....	1101
12.6.4. Get image member schema .....	1103
13. Image Service API v1 (SUPPORTED) .....	1104
13.1. API versions .....	1104
13.1.1. List API versions .....	1105
13.2. Images .....	1106
13.2.1. Create image .....	1107
13.2.2. List image details .....	1108
13.2.3. Show image details and image binary .....	1109
13.2.4. Update image .....	1110
13.2.5. Delete image .....	1111

13.3. Members .....	1111
13.3.1. Replace member .....	1112
13.3.2. Add member .....	1113
13.3.3. Delete image .....	1114
13.4. Shared images .....	1114
13.4.1. List shared images .....	1115
14. Networking API v2.0 (CURRENT) .....	1116
14.1. API versions .....	1116
14.1.1. List API versions .....	1117
14.1.2. Show API v2.0 details .....	1118
14.2. Networks .....	1119
14.2.1. List API versions .....	1121
14.2.2. Show API v2.0 details .....	1122
14.2.3. List extensions .....	1124
14.2.4. Get extension details .....	1130
14.2.5. List networks .....	1131
14.2.6. Create network .....	1133
14.2.7. Bulk create networks .....	1135
14.2.8. Show network .....	1137
14.2.9. Update network .....	1139
14.2.10. Delete network .....	1141
14.3. Subnets .....	1141
14.3.1. List subnets .....	1142
14.3.2. Create subnet .....	1144
14.3.3. Bulk create subnet .....	1146
14.3.4. Show subnet .....	1149
14.3.5. Update subnet .....	1151
14.3.6. Delete subnet .....	1153
14.4. Ports .....	1153
14.4.1. List ports .....	1154
14.4.2. Create port .....	1157
14.4.3. Bulk create ports .....	1159
14.4.4. Show port .....	1162
14.4.5. Update port .....	1164
14.4.6. Delete port .....	1167
15. Networking API v2.0 extensions .....	1168
15.1. Extensions .....	1168
15.1.1. List extensions .....	1169
15.1.2. Get extension details .....	1175
15.2. Quotas extension (quotas) .....	1175
15.2.1. List quotas .....	1177
15.2.2. Show quota .....	1178
15.2.3. Update quota .....	1179
15.2.4. Reset quota .....	1180
15.3. Networks provider extended attributes (networks) .....	1180
15.3.1. List networks .....	1181
15.3.2. Create network .....	1183
15.3.3. Show network details .....	1185
15.3.4. Update network .....	1187
15.3.5. Delete network .....	1189
15.4. Networks multiple provider extension (networks) .....	1189

15.4.1. List networks .....	1190
15.4.2. Create network with multiple segment mappings .....	1192
15.4.3. Show details for a network with multiple segments .....	1194
15.5. Ports binding extended attributes (ports) .....	1194
15.5.1. List ports .....	1196
15.5.2. Create port .....	1199
15.5.3. Show port .....	1201
15.5.4. Update port .....	1203
15.6. Security groups and rules (security-groups) .....	1204
15.6.1. List security groups .....	1205
15.6.2. Create security group .....	1207
15.6.3. Show security group .....	1209
15.6.4. Delete security group .....	1211
15.6.5. List security group rules .....	1212
15.6.6. Create security group rule .....	1214
15.6.7. Show security group rule .....	1215
15.6.8. Delete security group rule .....	1216
15.7. Layer-3 networking .....	1216
15.7.1. List routers .....	1218
15.7.2. Create router .....	1219
15.7.3. Show router details .....	1221
15.7.4. Update router .....	1222
15.7.5. Delete router .....	1223
15.7.6. Add interface to router .....	1224
15.7.7. Remove interface from router .....	1226
15.7.8. Create floating IP .....	1227
15.7.9. Show floating IP details .....	1229
15.7.10. Update floating IP .....	1230
15.7.11. Delete floating IP .....	1232
15.8. Metering labels and rules .....	1232
15.8.1. List metering labels .....	1233
15.8.2. Create metering label .....	1234
15.8.3. Show metering label .....	1236
15.8.4. Delete metering label .....	1237
15.8.5. List metering label rules .....	1238
15.8.6. Create metering label rule .....	1239
15.8.7. Show metering label rule .....	1241
15.8.8. Delete metering label rule .....	1243
15.9. Load-Balancer-as-a-Service (LBaaS) .....	1243
15.9.1. Create load balancer .....	1245
15.9.2. List load balancers .....	1246
15.9.3. Show load balancer configuration details .....	1247
15.9.4. Update load balancer .....	1248
15.9.5. Delete load balancer .....	1249
15.9.6. Create listener .....	1250
15.9.7. List listeners .....	1251
15.9.8. Show listener configuration details .....	1252
15.9.9. Update listener .....	1253
15.9.10. Delete listener .....	1254
15.9.11. Create load balancer pool .....	1255
15.9.12. List pools .....	1256

15.9.13. Show pool configuration details .....	1257
15.9.14. Update pool .....	1258
15.9.15. Delete pool .....	1259
15.9.16. Create load balancer member .....	1260
15.9.17. List pool members .....	1261
15.9.18. Show member configuration details .....	1262
15.9.19. Update member .....	1263
15.9.20. Delete member .....	1264
15.9.21. Create load balancer health monitor .....	1265
15.9.22. List health monitors .....	1266
15.9.23. Show health monitor configuration details .....	1267
15.9.24. Update health monitor .....	1268
15.9.25. Delete health monitor .....	1269
16. Object Storage API v1 (SUPPORTED) .....	1270
16.1. Accounts .....	1270
16.1.1. Show account details and list containers .....	1271
16.1.2. Create, update, or delete account metadata .....	1274
16.1.3. Show account metadata .....	1277
16.2. Containers .....	1278
16.2.1. Show container details and list objects .....	1280
16.2.2. Create container .....	1284
16.2.3. Delete container .....	1286
16.2.4. Create, update, or delete container metadata .....	1288
16.2.5. Show container metadata .....	1291
16.3. Objects .....	1292
16.3.1. Get object content and metadata .....	1294
16.3.2. Create or replace object .....	1298
16.3.3. Copy object .....	1301
16.3.4. Delete object .....	1304
16.3.5. Show object metadata .....	1306
16.3.6. Create or update object metadata .....	1309
17. Orchestration API v1 (CURRENT) .....	1312
17.1. API versions .....	1312
17.1.1. List versions .....	1313
17.2. Stacks .....	1313
17.2.1. Create stack .....	1314
17.2.2. Adopt stack .....	1315
17.2.3. List stack data .....	1316
17.2.4. Find stack .....	1318
17.2.5. Show stack details .....	1319
17.2.6. Update stack .....	1320
17.2.7. Delete stack .....	1321
17.2.8. Preview stack .....	1322
17.2.9. Abandon stack .....	1325
17.3. Stack actions .....	1326
17.3.1. Suspend stack .....	1327
17.3.2. Resume stack .....	1328
17.4. Stack resources .....	1328
17.4.1. Find stack resources .....	1329
17.4.2. List resources .....	1330
17.4.3. Show resource data .....	1331

17.4.4. Show resource metadata .....	1332
17.4.5. List resource types .....	1333
17.4.6. Show resource schema .....	1334
17.4.7. Show resource template .....	1335
17.5. Stack events .....	1335
17.5.1. Find stack events .....	1336
17.5.2. List stack events .....	1337
17.5.3. List resource events .....	1338
17.5.4. Show event details .....	1339
17.6. Templates .....	1339
17.6.1. Get stack template .....	1340
17.6.2. Validate template .....	1341
17.7. Build info .....	1341
17.7.1. Show build information .....	1342
17.8. Software configuration .....	1342
17.8.1. Create configuration .....	1344
17.8.2. Show configuration details .....	1346
17.8.3. Delete config .....	1348
17.8.4. List deployments .....	1349
17.8.5. Create deployment .....	1350
17.8.6. Show server configuration metadata .....	1351
17.8.7. Show deployment details .....	1354
17.8.8. Update deployment .....	1355
17.8.9. Delete deployment .....	1357
18. Telemetry API v2 (CURRENT) .....	1358
18.1. Alarms .....	1358
18.1.1. List alarms .....	1359
18.1.2. Create alarm .....	1361
18.1.3. Show alarm .....	1363
18.1.4. Update alarm .....	1365
18.1.5. Delete alarm .....	1367
18.1.6. Update alarm state .....	1368
18.1.7. Show alarm state .....	1369
18.1.8. Show alarm history .....	1370
18.2. Meters .....	1370
18.2.1. List meters .....	1371
18.2.2. Show meter .....	1372
18.2.3. Create meter .....	1374
18.2.4. Show meter statistics .....	1376
18.3. Resources .....	1377
18.3.1. List resources .....	1378
18.3.2. Show resource information .....	1380

# 1. Block Storage API v2 (CURRENT)

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

Method	URI	Description
API versions		
<b>GET</b>	/	Lists information about all Block Storage API versions.
<b>GET</b>	/v2	Shows details for Block Storage API v2.
API extensions		
<b>GET</b>	/v2/{tenant_id}/extensions	Lists Block Storage API extensions.
Volumes		
<b>POST</b>	/v2/{tenant_id}/volumes	Creates a volume.
<b>GET</b>	/v2/{tenant_id}/volumes	Lists summary information for all Block Storage volumes that the tenant who submits the request can access.
<b>GET</b>	/v2/{tenant_id}/volumes/detail	Lists detailed information for all Block Storage volumes that the tenant who submits the request can access.
<b>GET</b>	/v2/{tenant_id}/volumes/{volume_id}	Shows information about a specified volume.
<b>PUT</b>	/v2/{tenant_id}/volumes/{volume_id}{?display_description,display_name}	Updates a volume.
<b>DELETE</b>	/v2/{tenant_id}/volumes/{volume_id}	Deletes a specified volume.
Volume types		
<b>GET</b>	/v2/{tenant_id}/types	Lists volume types.
<b>POST</b>	/v2/{tenant_id}/types	Creates a volume type.
<b>GET</b>	/v2/{tenant_id}/types/{volume_type_id}	Shows information about a specified volume type.
<b>DELETE</b>	/v2/{tenant_id}/types/{volume_type_id}	Deletes a specified volume type.
Snapshots		
<b>POST</b>	/v2/{tenant_id}/snapshots{?snapshot,volume_id,force,display_name,display_description}	Creates a snapshot, which is a point-in-time copy of a volume. You can create a volume from the snapshot.
<b>GET</b>	/v2/{tenant_id}/snapshots	Lists summary information for all Block Storage snapshots that the tenant who submits the request can access.
<b>GET</b>	/v2/{tenant_id}/snapshots/detail	Lists detailed information for all Block Storage snapshots that the tenant who submits the request can access.
<b>GET</b>	/v2/{tenant_id}/snapshots/{snapshot_id}	Shows information for a specified snapshot.
<b>PUT</b>	/v2/{tenant_id}/snapshots/{snapshot_id}{?display_description,display_name}	Updates a specified snapshot.
<b>DELETE</b>	/v2/{tenant_id}/snapshots/{snapshot_id}	Deletes a specified snapshot.
<b>GET</b>	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Shows the metadata for a specified snapshot.
<b>PUT</b>	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Updates the metadata for a specified snapshot.
Quality of service (QoS) specifications (qos-specs)		

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/qos-specs	Creates a QoS specification.
<b>GET</b>	/v2/{tenant_id}/qos-specs	Lists quality of service (QoS) specifications.
<b>GET</b>	/v2/{tenant_id}/qos-specs/{qos_id}	Shows details for a specified QoS specification.
<b>DELETE</b>	/v2/{tenant_id}/qos-specs/{qos_id}	Deletes a specified QoS specification.
<b>GET</b>	/v2/{tenant_id}/qos-specs/{qos_id}/associate{?vol_type_id}	Associates a QoS specification with a specified volume type.
<b>GET</b>	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate{?vol_type_id}	Disassociates a QoS specification from a specified volume type.
<b>GET</b>	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate_all	Disassociates a specified QoS specification from all associations.
<b>GET</b>	/v2/{tenant_id}/qos-specs/{qos_id}/associations	Gets all associations for a specified QoS specification.
Quota sets extension (os-quota-sets)		
<b>GET</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Shows quotas for a tenant.
<b>PUT</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.
<b>DELETE</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Deletes quotas for a tenant so the quotas revert to default values.
<b>GET</b>	/v2/{tenant_id}/os-quota-sets/defaults	Gets default quotas for a tenant.
<b>GET</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a specified tenant and user.
<b>POST</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a specified tenant/project and user.
<b>DELETE</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Deletes quotas for a user so that the quotas revert to default values.
<b>GET</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}	Shows details for quotas for a specified tenant and user.
Limits extension (limits)		
<b>GET</b>	/v2/{tenant_id}/limits	Shows absolute limits for a tenant.

## 1.1. API versions

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

## 1.1.1. List API versions

Method	URI	Description
GET	/	Lists information about 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": [
        {
            "status": "CURRENT",
            "updated": "2012-01-04T11:33:21Z",
            "id": "v1.0",
            "links": [
                {
                    "href": "http://23.253.228.211:8776/v1/",
                    "rel": "self"
                }
            ]
        },
        {
            "status": "CURRENT",
            "updated": "2012-11-21T11:33:21Z",
            "id": "v2.0",
            "links": [
                {
                    "href": "http://23.253.228.211:8776/v2/",
                    "rel": "self"
                }
            ]
        }
    ]
}
```

## 1.1.2. Show API version details

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

**Normal response codes:** 200, 203

### 1.1.2.1. Request

This operation does not accept a request body.

### 1.1.2.2. Response

#### Example 1.2. Show API version details: JSON response

```
{
  "version": {
    "status": "CURRENT",
    "updated": "2012-01-04T11:33:21Z",
    "media-types": [
      {
        "base": "application/xml",
        "type": "application/vnd.openstack.volume+xml;version=1"
      },
      {
        "base": "application/json",
        "type": "application/vnd.openstack.volume+json;version=1"
      }
    ],
    "id": "v1.0",
    "links": [
      {
        "href": "http://23.253.228.211:8776/v2/",
        "rel": "self"
      },
      {
        "href": "http://jorgew.github.com/block-storage-api/content/os-block-storage-1.0.pdf",
        "type": "application/pdf",
        "rel": "describedby"
      },
      {
        "href": "http://docs.rackspacecloud.com/servers/api/v1.1/application.wadl",
        "type": "application/vnd.sun.wadl+xml",
        "rel": "describedby"
      }
    ]
  }
}
```

## 1.2. API 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 1.3. 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 1.4. 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"
               name="TypesManager">
        <description>Manage volume types.</description>
    </extension>

```

```
        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 backend 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. Volumes

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/volumes	Creates a volume.
<b>GET</b>	/v2/{tenant_id}/volumes	Lists summary information for all Block Storage volumes that the tenant who submits the request can access.
<b>GET</b>	/v2/{tenant_id}/volumes/detail	Lists detailed information for all Block Storage volumes that the tenant who submits the request can access.
<b>GET</b>	/v2/{tenant_id}/volumes/{volume_id}	Shows information about a specified volume.
<b>PUT</b>	/v2/{tenant_id}/volumes/{volume_id}{{?display_description,display_name}}	Updates a volume.
<b>DELETE</b>	/v2/{tenant_id}/volumes/{volume_id}	Deletes a specified volume.

## 1.3.1. Create volume

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

To create a bootable volume, include the image ID and set the `bootable` flag to `true` in the request body.

**Normal response codes:** 202

### 1.3.1.1. 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 1.5. Create volume: JSON request

```
{
    "volume": {
        "availability_zone": null,
        "source_volid": null,
        "display_description": null,
        "snapshot_id": null,
        "size": 10,
        "display_name": "my_volume",
        "imageRef": null,
        "volume_type": 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.
display_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.
size	Int <i>(Required)</i>	The size of the volume, in GBs.
display_name	String <i>(Optional)</i>	The volume name.
imageRef	Uuid	The ID of the image from which you want to create the volume. Required to create a bootable volume.

Name	Type	Description
	(Optional)	
volume_type	String	The associated volume type.
	(Optional)	
bootable	Boolean	Enables or disables the bootable attribute. You can boot an instance from a bootable volume.
	(Optional)	
metadata	String	One or more metadata key and value pairs to associate with the volume.
	(Optional)	

### Example 1.6. Create volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume
  xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
  display_name="vol-001" display_description="Another volume."
  size="2"/>
```

This operation does not accept a request body.

### 1.3.1.2. Response

#### Example 1.7. Create volume: JSON response

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

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

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

Name	Type	Description
bootable	Boolean <i>(Required)</i>	Enables or disables the bootable attribute. You can boot an instance from a bootable volume.
created_at	Datetime <i>(Required)</i>	Date and time when the volume was created.
display_description	String <i>(Required)</i>	The volume description.
volume_type	String <i>(Required)</i>	The associated volume type.
snapshot_id	Uuid <i>(Required)</i>	To create a volume from an existing volume snapshot, specify the ID of the existing volume snapshot.
source_volid	Uuid <i>(Required)</i>	To create a volume from an existing volume, specify the ID of the existing volume.
metadata	String <i>(Required)</i>	One or more metadata key and value pairs to associate with the volume.
id	Uuid <i>(Required)</i>	The volume ID.
size	Int <i>(Required)</i>	The size of the volume, in GBs.

### Example 1.8. 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"
         display_name="vol-001" availability_zone="nova" bootable="false"
         created_at="2014-02-21 20:18:33.122452"
         display_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.

## 1.3.2. List volumes

Method	URI	Description
GET	/v2/{tenant_id}/volumes	Lists summary information for all Block Storage volumes that the tenant who submits the request can access.

**Normal response codes:** 200

### 1.3.2.1. 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 operation does not accept a request body.

### 1.3.2.2. Response

#### Example 1.9. 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 1.10. 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.

## 1.3.3. List volumes (detailed)

Method	URI	Description
GET	/v2/{tenant_id}/volumes/detail	Lists detailed information for all Block Storage volumes that the tenant who submits the request can access.

**Normal response codes:** 200

### 1.3.3.1. Request

This table shows the URI parameters for the list volumes (detailed) request:

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

This operation does not accept a request body.

### 1.3.3.2. Response

#### Example 1.11. List volumes (detailed): 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 1.12. List volumes (detailed): 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,
            "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,
        "metadata": {
            "contents": "not junk"
        }
    }
]
```

This operation does not return a response body.

## 1.3.4. Show volume information

Method	URI	Description
GET	/v2/{tenant_id}/volumes/{volume_id}	Shows information about a specified volume.

**Normal response codes:** 200

### 1.3.4.1. Request

This table shows the URI parameters for the show volume information 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.

### 1.3.4.2. Response

#### Example 1.13. Show volume information: 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_valid="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 1.14. Show volume information: 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,
"metadata": {
    "contents": "not junk"
}
}
```

This operation does not return a response body.

## 1.3.5. Update volume

Method	URI	Description
PUT	/v2/{tenant_id}/volumes/{volume_id}{{?display_description,display_name}}	Updates a volume.

**Normal response codes:** 200

### 1.3.5.1. 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.

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

Name	Type	Description
display_description	String <i>(Optional)</i>	A description of the volume.
display_name	String <i>(Optional)</i>	The name of the volume.

#### Example 1.15. Update volume: XML request

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

#### Example 1.16. Update volume: JSON request

```
{
    "volume": {
        "display_name": "vol-003",
        "display_description": "This is yet, another volume."
    }
}
```

This operation does not accept a request body.

### 1.3.5.2. Response

#### Example 1.17. 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" display_name="vol-003" availability_zone="nova"
```

```
created_at="2013-02-25 02:40:21"
display_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 1.18. 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",
        "display_description": "This is yet, another volume.",
        "display_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.

## 1.3.6. Delete volume

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/volumes/{volume_id}	Deletes a specified volume.

**Normal response codes:** 202

### 1.3.6.1. 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.

## 1.4. Volume types

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/types	Lists volume types.
<b>POST</b>	/v2/{tenant_id}/types	Creates a volume type.
<b>GET</b>	/v2/{tenant_id}/types/{volume_type_id}	Shows information about a specified volume type.
<b>DELETE</b>	/v2/{tenant_id}/types/{volume_type_id}	Deletes a specified volume type.

## 1.4.1. List volume types

Method	URI	Description
GET	/v2/{tenant_id}/types	Lists volume types.

**Normal response codes:** 200

### 1.4.1.1. 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.

### 1.4.1.2. Response

#### Example 1.19. 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 1.20. 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.

## 1.4.2. Create volume type

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

**Normal response codes:** 200

### 1.4.2.1. 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 1.21. 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 1.22. Create volume type: JSON request

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

This operation does not accept a request body.

### 1.4.2.2. Response

#### Example 1.23. 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 1.24. 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.

## 1.4.3. Show volume type information

Method	URI	Description
GET	/v2/{tenant_id}/types/{volume_type_id}	Shows information about a specified volume type.

**Normal response codes:** 200

### 1.4.3.1. Request

This table shows the URI parameters for the show volume type information 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.

### 1.4.3.2. Response

#### Example 1.25. Show volume type information: JSON response

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

#### Example 1.26. Show volume type information: 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.

## 1.4.4. Delete volume type

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

**Normal response codes:** 202

### 1.4.4.1. 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.

## 1.5. Snapshots

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/snapshots{?snapshot,volume_id,force,display_name,display_description}	Creates a snapshot, which is a point-in-time copy of a volume. You can create a volume from the snapshot.
<b>GET</b>	/v2/{tenant_id}/snapshots	Lists summary information for all Block Storage snapshots that the tenant who submits the request can access.
<b>GET</b>	/v2/{tenant_id}/snapshots/detail	Lists detailed information for all Block Storage snapshots that the tenant who submits the request can access.
<b>GET</b>	/v2/{tenant_id}/snapshots/{snapshot_id}	Shows information for a specified snapshot.
<b>PUT</b>	/v2/{tenant_id}/snapshots/{snapshot_id}{?display_description,display_name}	Updates a specified snapshot.
<b>DELETE</b>	/v2/{tenant_id}/snapshots/{snapshot_id}	Deletes a specified snapshot.
<b>GET</b>	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Shows the metadata for a specified snapshot.
<b>PUT</b>	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Updates the metadata for a specified snapshot.

## 1.5.1. Create snapshot

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

Normal response codes: 202

### 1.5.1.1. 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.
display_name	String <i>(Optional)</i>	Name of the snapshot. Default==None.
display_description	String <i>(Optional)</i>	Description of snapshot. Default==None.

### Example 1.27. 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 1.28. 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.

### 1.5.1.2. Response

#### Example 1.29. 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 1.30. 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.

## 1.5.2. List snapshots

Method	URI	Description
GET	/v2/{tenant_id}/snapshots	Lists summary information for all Block Storage snapshots that the tenant who submits the request can access.

**Normal response codes:** 200

### 1.5.2.1. 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 operation does not accept a request body.

### 1.5.2.2. Response

#### Example 1.31. 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 1.32. 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.

## 1.5.3. List snapshots (detailed)

Method	URI	Description
GET	/v2/{tenant_id}/snapshots/detail	Lists detailed information for all Block Storage snapshots that the tenant who submits the request can access.

**Normal response codes:** 200

### 1.5.3.1. Request

This table shows the URI parameters for the list snapshots (detailed) request:

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

This operation does not accept a request body.

### 1.5.3.2. Response

#### Example 1.33. List snapshots (detailed): 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 1.34. List snapshots (detailed): 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.

## 1.5.4. Show snapshot information

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

**Normal response codes:** 200

### 1.5.4.1. Request

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

Name	Type	Description
{tenant_id}	String	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.

### 1.5.4.2. Response

#### Example 1.35. Show snapshot information: 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 1.36. Show snapshot information: 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.

## 1.5.5. Update snapshot

Method	URI	Description
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}{?display_description,display_name}	Updates a specified snapshot.

**Normal response codes:** 200

### 1.5.5.1. 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 unique identifier of an existing snapshot.

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

Name	Type	Description
display_description	String <i>(Optional)</i>	Describes the snapshot.
display_name	String <i>(Optional)</i>	The name of the snapshot.

#### Example 1.37. Update snapshot: XML request

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

#### Example 1.38. Update snapshot: JSON request

```
{
    "snapshot": {
        "display_name": "snap-002",
        "display_description": "This is yet, another snapshot."
    }
}
```

This operation does not accept a request body.

### 1.5.5.2. Response

#### Example 1.39. 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"
display_description="This is yet, another snapshot"
created_at="2013-02-20T08:11:34.000000"
volume_id="2402b902-0b7a-458c-9c07-7435a826f794"
size="1"
id="4b502fcb-1f26-45f8-9fe5-3b9a0a52eaf2"
display_name="vol-002"
os-extended-snapshot-attributes:project_id=
"0c2eba2c5af04d3f9e9d0d410b371fde"
os-extended-snapshot-attributes:progress="100%">
<metadata/>
</snapshot>
```

### Example 1.40. Update snapshot: JSON response

```
{
  "snapshot": {
    "created_at": "2013-02-20T08:11:34.000000",
    "display_description": "This is yet, another snapshot",
    "display_name": "vol-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.

## 1.5.6. Delete snapshot

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

**Normal response codes:** 202

### 1.5.6.1. 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 unique identifier of an existing snapshot.

This operation does not accept a request body.

## 1.5.7. Show snapshot metadata

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

**Normal response codes:** 200

### 1.5.7.1. 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 unique identifier of an existing snapshot.

This operation does not accept a request body.

### 1.5.7.2. Response

#### Example 1.41. 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 1.42. 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.

## 1.5.8. Update snapshot metadata

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

**Normal response codes:** 200

### 1.5.8.1. 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 unique identifier of an existing snapshot.

#### Example 1.43. Update snapshot metadata: XML request

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

#### Example 1.44. Update snapshot metadata: JSON request

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

This operation does not accept a request body.

### 1.5.8.2. Response

#### Example 1.45. 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 1.46. Update snapshot metadata: JSON response

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

This operation does not return a response body.

## 1.6. Quality of service (QoS) specifications (qos-specs)

Administrators only, depending on policy settings. Create, list, show details for, associate, disassociate, and delete quality of service (QoS) specifications.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/qos-specs	Creates a QoS specification.
<b>GET</b>	/v2/{tenant_id}/qos-specs	Lists quality of service (QoS) specifications.
<b>GET</b>	/v2/{tenant_id}/qos-specs/{qos_id}	Shows details for a specified QoS specification.
<b>DELETE</b>	/v2/{tenant_id}/qos-specs/{qos_id}	Deletes a specified QoS specification.
<b>GET</b>	/v2/{tenant_id}/qos-specs/{qos_id}/associate{?vol_type_id}	Associates a QoS specification with a specified volume type.
<b>GET</b>	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate{?vol_type_id}	Disassociates a QoS specification from a specified volume type.
<b>GET</b>	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate_all	Disassociates a specified QoS specification from all associations.
<b>GET</b>	/v2/{tenant_id}/qos-specs/{qos_id}/associations	Gets all associations for a specified QoS specification.

## 1.6.1. Create QoS specification

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

**Normal response codes:** 202

### 1.6.1.1. 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 1.47. 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 1.48. 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.

### 1.6.1.2. Response

#### Example 1.49. 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 1.50. 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.

## 1.6.2. List QoS specs

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

**Normal response codes:** 200, 300

### 1.6.2.1. 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.

### 1.6.2.2. Response

#### Example 1.51. 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 1.52. 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.

## 1.6.3. Show QoS specification details

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

**Normal response codes:** 200

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

### 1.6.3.1. 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.

### 1.6.3.2. Response

#### Example 1.53. 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 1.54. 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.

## 1.6.4. Delete QoS specification

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

**Normal response codes:** 202

### 1.6.4.1. 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 the specified QoS specification even if it is in-use.

This operation does not accept a request body.

## 1.6.5. 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 specified volume type.

**Normal response codes:** 202

### 1.6.5.1. 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.

## 1.6.6. Disassociate QoS specification from volume type

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate{?vol_type_id}	Disassociates a QoS specification from a specified volume type.

**Normal response codes:** 202

### 1.6.6.1. 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.

## 1.6.7. Disassociate QoS specification from all associations

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

**Normal response codes:** 202

### 1.6.7.1. 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.

## 1.6.8. Get all associations for QoS specification

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

Normal response codes: 200

### 1.6.8.1. 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.

### 1.6.8.2. Response

#### Example 1.55. 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 1.56. 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.

## 1.7. Quota sets extension (os-quota-sets)

Administrators only, depending on policy settings. View, update, and delete quotas for a tenant.

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

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

## 1.7.1. Show quotas

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

Normal response codes: 200

### 1.7.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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.

### 1.7.1.2. Response

#### Example 1.57. 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 <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	The number of floating IP addresses allowed for each tenant.

Name	Type	Description
	(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 1.58. 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.

## 1.7.2. Update quotas

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

Normal response codes: 200

### 1.7.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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 1.59. 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 <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 1.60. 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.

### 1.7.2.2. Response

#### Example 1.61. 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 <i>(Required)</i>	String	A quota_set object.
cores <i>(Required)</i>	Int	The number of instance cores allowed for each tenant.
fixed_ips <i>(Required)</i>	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
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 1.62. 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.

## 1.7.3. Delete quotas

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Deletes quotas for a tenant so the quotas revert to default values.

**Normal response codes:** 200

### 1.7.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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.

## 1.7.4. Get default quotas

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

Normal response codes: 200

### 1.7.4.1. Request

This table shows the URI parameters for the get default quotas 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.

### 1.7.4.2. Response

#### Example 1.63. 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 1.64. 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.

## 1.7.5. 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 specified tenant and user.

Normal response codes: 200

### 1.7.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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 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.

### 1.7.5.2. Response

#### Example 1.65. 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 1.66. 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.

## 1.7.6. Update quotas for user

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

Normal response codes: 200

### 1.7.6.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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 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 1.67. 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 1.68. 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.

### 1.7.6.2. Response

### Example 1.69. 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 1.70. 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.

## 1.7.7. Delete quotas for user

Method	URI	Description
<b>DELETE</b>	/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

### 1.7.7.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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 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.

## 1.7.8. 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 specified tenant and user.

**Normal response codes:** 200

### 1.7.8.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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 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.

### 1.7.8.2. Response

#### Example 1.71. 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
    }
}
```

## 1.8. Limits extension (limits)

Show absolute limits for a tenant.

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

## 1.8.1. Show absolute limits

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

**Normal response codes:** 200, 203

### 1.8.1.1. 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.

### 1.8.1.2. Response

#### Example 1.72. Show absolute limits: JSON response

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

#### Example 1.73. 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.

## 2. Block Storage API v1 (CURRENT)

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

Method	URI	Description
API versions		
<b>GET</b>	/	Lists information about all Block Storage API versions.
<b>GET</b>	/v1	Shows details for Block Storage API v1.
Volumes		
<b>POST</b>	/v1/{tenant_id}/volumes	Creates a volume.
<b>GET</b>	/v1/{tenant_id}/volumes	Lists simple volume entities.
<b>GET</b>	/v1/{tenant_id}/volumes/detail	Lists details for volume entities.
<b>GET</b>	/v1/{tenant_id}/volumes/{volume_id}	Shows information about a specified volume.
<b>DELETE</b>	/v1/{tenant_id}/volumes/{volume_id}	Deletes a specified volume.
Volume types		
<b>GET</b>	/v1/{tenant_id}/types	Request a list of volume types.
<b>POST</b>	/v1/{tenant_id}/types	Creates a volume type.
<b>GET</b>	/v1/{tenant_id}/types/{volume_type_id}	Shows information about a specified volume type.
<b>DELETE</b>	/v1/{tenant_id}/types/{volume_type_id}	Deletes a specified volume type.
Snapshots		
<b>POST</b>	/v1/{tenant_id}/snapshots	Creates a snapshot.
<b>GET</b>	/v1/{tenant_id}/snapshots	Lists simple snapshot entities.
<b>GET</b>	/v1/{tenant_id}/snapshots/detail	Lists details for snapshot entities.
<b>GET</b>	/v1/{tenant_id}/snapshots/{snapshot_id}	Shows information about a specified snapshot.
<b>DELETE</b>	/v1/{tenant_id}/snapshots/{snapshot_id}	Deletes a specified snapshot.
<b>GET</b>	/v1/{tenant_id}/snapshots/{snapshot_id}/metadata	Shows the metadata for a specified snapshot.
<b>PUT</b>	/v1/{tenant_id}/snapshots/{snapshot_id}/metadata	Updates the metadata for a specified snapshot.

### 2.1. API versions

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

## 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": [
        {
            "status": "CURRENT",
            "updated": "2012-01-04T11:33:21Z",
            "id": "v1.0",
            "links": [
                {
                    "href": "http://23.253.228.211:8776/v1/",
                    "rel": "self"
                }
            ]
        },
        {
            "status": "CURRENT",
            "updated": "2012-11-21T11:33:21Z",
            "id": "v2.0",
            "links": [
                {
                    "href": "http://23.253.228.211:8776/v2/",
                    "rel": "self"
                }
            ]
        }
    ]
}
```

## 2.1.2. Show API v1 details

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

**Normal response codes:** 200, 203

### 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": {
    "status": "CURRENT",
    "updated": "2012-01-04T11:33:21Z",
    "media-types": [
      {
        "base": "application/xml",
        "type": "application/vnd.openstack.volume+xml;version=1"
      },
      {
        "base": "application/json",
        "type": "application/vnd.openstack.volume+json;version=1"
      }
    ],
    "id": "v1.0",
    "links": [
      {
        "href": "http://23.253.228.211:8776/v1/",
        "rel": "self"
      },
      {
        "href": "http://jorgew.github.com/block-storage-api/content/os-block-storage-1.0.pdf",
        "type": "application/pdf",
        "rel": "describedby"
      },
      {
        "href": "http://docs.rackspacecloud.com/servers/api/v1.1/application.wadl",
        "type": "application/vnd.sun.wadl+xml",
        "rel": "describedby"
      }
    ]
  }
}
```

## 2.2. Volumes

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

Method	URI	Description
<b>GET</b>	/v1/{tenant_id}/volumes	Lists simple volume entities.
<b>GET</b>	/v1/{tenant_id}/volumes/detail	Lists details for volume entities.
<b>GET</b>	/v1/{tenant_id}/volumes/{volume_id}	Shows information about a specified volume.
<b>DELETE</b>	/v1/{tenant_id}/volumes/{volume_id}	Deletes a specified 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 2.3. Create volume: XML request

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

#### Example 2.4. Create volume: JSON request

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

This operation does not accept a request body.

### 2.2.1.2. Response

#### Example 2.5. Create volume: XML response

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

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

### Example 2.6. Create volume: JSON response

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

**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 2.7. List volumes: XML response

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

#### Example 2.8. List volumes: JSON response

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

This operation does not return a response body.

## 2.2.3. List volumes (detailed)

Method	URI	Description
GET	/v1/{tenant_id}/volumes/detail	Lists details for volume entities.

**Normal response codes:** 200

### 2.2.3.1. Request

This table shows the URI parameters for the list volumes (detailed) 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 2.9. List volumes (detailed): XML response

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

#### Example 2.10. List volumes (detailed): 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"
    }
]
```

This operation does not return a response body.

## 2.2.4. Show volume information

Method	URI	Description
GET	/v1/{tenant_id}/volumes/{volume_id}	Shows information about a specified volume.

**Normal response codes:** 200

### 2.2.4.1. Request

This table shows the URI parameters for the show volume information 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 2.11. Show volume information: XML response

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

#### Example 2.12. Show volume information: JSON response

```
{
    "volume": {
        "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",
        "bootable": "false",
        "snapshot_id": null,
        "attachments": [],
        "created_at": "2012-02-14T20:53:07Z"
    }
}
```

```
    }  
}
```

This operation does not return a response body.

## 2.2.5. Delete volume

Method	URI	Description
<b>DELETE</b>	/v1/{tenant_id}/volumes/{volume_id}	Deletes a specified 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
<b>GET</b>	/v1/{tenant_id}/types	Request a list of volume types.
<b>POST</b>	/v1/{tenant_id}/types	Creates a volume type.
<b>GET</b>	/v1/{tenant_id}/types/{volume_type_id}	Shows information about a specified volume type.
<b>DELETE</b>	/v1/{tenant_id}/types/{volume_type_id}	Deletes a specified volume type.

## 2.3.1. List volume types

Method	URI	Description
GET	/v1/{tenant_id}/types	Request a list of 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 2.13. List volume types: XML response

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

#### Example 2.14. List volume types: JSON response

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

This operation does not return a response body.

## 2.3.2. Create volume type

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

**Normal response codes:** 200

### 2.3.2.1. Request

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

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

#### Example 2.15. Create volume type: XML request

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

#### Example 2.16. Create volume type: JSON request

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

This operation does not accept a request body.

### 2.3.2.2. Response

#### Example 2.17. Create volume type: XML response

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

#### Example 2.18. Create volume type: JSON response

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

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

This operation does not return a response body.

## 2.3.3. Show volume type information

Method	URI	Description
GET	/v1/{tenant_id}/types/{volume_type_id}	Shows information about a specified volume type.

**Normal response codes:** 200

### 2.3.3.1. Request

This table shows the URI parameters for the show volume type information 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 2.19. Show volume type information: 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 2.20. Show volume type information: 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
<b>DELETE</b>	/v1/{tenant_id}/types/{volume_type_id}	Deletes a specified 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
<b>POST</b>	/v1/{tenant_id}/snapshots	Creates a snapshot.
<b>GET</b>	/v1/{tenant_id}/snapshots	Lists simple snapshot entities.
<b>GET</b>	/v1/{tenant_id}/snapshots/detail	Lists details for snapshot entities.
<b>GET</b>	/v1/{tenant_id}/snapshots/{snapshot_id}	Shows information about a specified snapshot.
<b>DELETE</b>	/v1/{tenant_id}/snapshots/{snapshot_id}	Deletes a specified snapshot.
<b>GET</b>	/v1/{tenant_id}/snapshots/{snapshot_id}/metadata	Shows the metadata for a specified snapshot.
<b>PUT</b>	/v1/{tenant_id}/snapshots/{snapshot_id}/metadata	Updates the metadata for a specified 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 2.21. 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 2.22. 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 2.23. 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 2.24. 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 snapshot entities.

**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 2.25. 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" />
    <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 2.26. 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"
        },
        {
            "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"
    }
]
```

This operation does not return a response body.

## 2.4.3. List snapshots (detailed)

Method	URI	Description
GET	/v1/{tenant_id}/snapshots/detail	Lists details for snapshot entities.

**Normal response codes:** 200

### 2.4.3.1. Request

This table shows the URI parameters for the list snapshots (detailed) 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 2.27. List snapshots (detailed): 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" />
    <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 2.28. List snapshots (detailed): 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"
        },
        {
            "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"
    }
]
```

This operation does not return a response body.

## 2.4.4. Show snapshot information

Method	URI	Description
GET	/v1/{tenant_id}/snapshots/{snapshot_id}	Shows information about a specified snapshot.

**Normal response codes:** 200

### 2.4.4.1. Request

This table shows the URI parameters for the show snapshot information 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 2.29. Show snapshot information: 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 2.30. Show snapshot information: 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 specified 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 the metadata for a specified 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 2.31. 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 2.32. 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 the metadata for a specified 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 2.33. Update snapshot metadata: XML request

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

#### Example 2.34. Update snapshot metadata: JSON request

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

This operation does not accept a request body.

### 2.4.7.2. Response

#### Example 2.35. 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 2.36. Update snapshot metadata: JSON response

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

This operation does not return a response body.

## 3. Compute API v2 (CURRENT)

Launch virtual machines from images or images stored on persistent volumes. API v1.1 is identical to API v2.



### Note

XML support in requests and responses has been deprecated for the Compute API v2.

Method	URI	Description
API versions		
<b>GET</b>	/	Lists information about all Compute API versions.
<b>GET</b>	/v2	Shows details for Compute API v2.
Limits		
<b>GET</b>	/v2/{tenant_id}/limits	Lists the current limits for the account.
Extensions		
<b>GET</b>	/v2/{tenant_id}/extensions	Lists available extensions.
<b>GET</b>	/v2/{tenant_id}/extensions/{alias}	Gets details about the specified extension.
Servers		
<b>GET</b>	/v2/{tenant_id}/servers{?changes-since,image,flavor,name,marker,limit,status,host}	Lists IDs, names, and links for all servers.
<b>POST</b>	/v2/{tenant_id}/servers	Creates a server.
<b>GET</b>	/v2/{tenant_id}/servers/detail{?changes-since,image,flavor,name,marker,limit,status,host}	Lists details for all servers.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}	Gets details for a specified server.
<b>PUT</b>	/v2/{tenant_id}/servers/{server_id}	Updates the editable attributes of the specified server.
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}	Deletes a specified server.
Server metadata		
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/metadata	Shows metadata for a specified server.
<b>PUT</b>	/v2/{tenant_id}/servers/{server_id}/metadata	Creates or replaces metadata for a specified server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/metadata	Updates metadata items by key for a specified server.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/metadata/{key}	Shows details for a metadata item by key for a specified server.
<b>PUT</b>	/v2/{tenant_id}/servers/{server_id}/metadata/{key}	Sets a metadata item by key for a specified server.
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}/metadata/{key}	Deletes a metadata item by key for a specified server.
Server addresses		
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/ips	Lists networks and addresses for a specified tenant and server.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/ips/{network_label}	Lists addresses for a specified tenant, server, and network.

Method	URI	Description
Server actions		
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Changes the password for a server. Specify the <code>changePassword</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Reboots the specified server. Specify the <code>reboot</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Rebuilds the specified server. Specify the <code>rebuild</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Resizes the specified server. Specify the <code>resize</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Confirms a pending resize action. Specify the <code>confirmResize</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Cancels and reverts a pending resize action. Specify the <code>revertResize</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Creates a new image. Specify the <code>createImage</code> action in the request body.
Flavors		
<b>GET</b>	/v2/{tenant_id}/flavors{?minDisk,minRam,marker,limit}	Lists IDs, names, and links for available flavors.
<b>GET</b>	/v2/{tenant_id}/flavors/detail{?minDisk,minRam,marker,limit}	Lists all details for available flavors.
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor.
Images		
<b>GET</b>	/v2/{tenant_id}/images{?changes-since,server,name,status,marker,limit,type}	Lists IDs, names, and links for available images.
<b>GET</b>	/v2/{tenant_id}/images/detail{?changes-since,server,name,status,marker,limit,type}	Lists all details for available images.
<b>GET</b>	/v2/{tenant_id}/images/{image_id}	Gets details for a specified image.
<b>DELETE</b>	/v2/{tenant_id}/images/{image_id}	Deletes a specified image.
Image metadata		
<b>GET</b>	/v2/{tenant_id}/images/{image_id}/metadata	Shows metadata for a specified image.
<b>PUT</b>	/v2/{tenant_id}/images/{image_id}/metadata	Creates or replaces metadata for a specified image.
<b>POST</b>	/v2/{tenant_id}/images/{image_id}/metadata	Updates metadata items by key for a specified image.
<b>GET</b>	/v2/{tenant_id}/images/{image_id}/metadata/{key}	Shows details for a metadata item by key for a specified image.
<b>PUT</b>	/v2/{tenant_id}/images/{image_id}/metadata/{key}	Creates or updates a metadata item by key for a specified image.
<b>DELETE</b>	/v2/{tenant_id}/images/{image_id}/metadata/{key}	Deletes a metadata item by key for a specified image.

## 3.1. API versions

List information for all API versions and show details about API v2.

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

Method	URI	Description
<b>GET</b>	/v2	Shows details for Compute API v2.

### 3.1.1. List API versions

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

**Normal response codes:** 200, 300

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

#### 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": [
    {
      "status": "CURRENT",
      "updated": "2011-01-21T11:33:21Z",
      "id": "v2.0",
      "links": [
        {
          "href": "http://23.253.228.211:8774/v2/",
          "rel": "self"
        }
      ]
    },
    {
      "status": "EXPERIMENTAL",
      "updated": "2013-07-23T11:33:21Z",
      "id": "v3.0",
      "links": [
        {
          "href": "http://23.253.228.211:8774/v3/",
          "rel": "self"
        }
      ]
    }
  ]
}
```

##### Example 3.2. List API versions: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<choices xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/common/api/v1.0">
  <version status="EXPERIMENTAL" id="v3.0">
    <media-types>
      <media-type base="application/json"
        type="application/vnd.openstack.compute+json;version=3"
```

```
        />
    </media-types>
    <atom:link href="http://23.253.228.211:8774/v3/.xml"
               rel="self"/>
</version>
<version status="CURRENT" id="v2.0">
    <media-types>
        <media-type base="application/xml"
                    type="application/vnd.openstack.compute+xml;version=2"/>
        <media-type base="application/json"
                    type="application/vnd.openstack.compute+json;version=2"
                    />
    </media-types>
    <atom:link href="http://23.253.228.211:8774/v2/.xml"
               rel="self"/>
</version>
</choices>
```

This operation does not return a response body.

### 3.1.2. Show API version details

Method	URI	Description
GET	/v2	Shows details for Compute API v2.

**Normal response codes:** 200, 203

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

#### 3.1.2.1. Request

This operation does not accept a request body.

#### 3.1.2.2. Response

##### Example 3.3. Show API version details: JSON response

```
{
  "version": {
    "status": "CURRENT",
    "updated": "2011-01-21T11:33:21Z",
    "media-types": [
      {
        "base": "application/xml",
        "type": "application/vnd.openstack.compute+xml;version=2"
      },
      {
        "base": "application/json",
        "type": "application/vnd.openstack.compute+json;version=2"
      }
    ],
    "id": "v2.0",
    "links": [
      {
        "href": "http://23.253.228.211:8774/v2/",
        "rel": "self"
      },
      {
        "href": "http://docs.openstack.org/api/openstack-compute/2/os-compute-devguide-2.pdf",
        "type": "application/pdf",
        "rel": "describedby"
      },
      {
        "href": "http://docs.openstack.org/api/openstack-compute/2/wadl/os-compute-2.wadl",
        "type": "application/vnd.sun.wadl+xml",
        "rel": "describedby"
      }
    ]
  }
}
```

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

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

### Example 3.4. Show API version details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<version xmlns:atom="http://www.w3.org/2005/Atom"
          xmlns="http://docs.openstack.org/common/api/v1.0" status="CURRENT"
          updated="2011-01-21T11:33:21Z" id="v2.0">
    <media-types>
        <media-type base="application/xml"
                    type="application/vnd.openstack.compute+xml;version=2" />
        <media-type base="application/json"
                    type="application/vnd.openstack.compute+json;version=2" />
    </media-types>
    <atom:link href="http://23.253.228.211:8774/v2/" rel="self"/>
    <atom:link
        href="http://docs.openstack.org/api/openstack-compute/2/os-compute-
devguide-2.pdf"
        type="application/pdf" rel="describedby" />
    <atom:link
        href="http://docs.openstack.org/api/openstack-compute/2/wadl/os-
compute-2.wadl"
        type="application/vnd.sun.wadl+xml" rel="describedby" />
</version>
```

## 3.2. Limits

Get rate and absolute limits.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/limits	Lists the current limits for the account.

## 3.2.1. List limits

Method	URI	Description
GET	/v2/{tenant_id}/limits	Lists the current limits for the account.

**Normal response codes:** 200, 203

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

### 3.2.1.1. Request

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

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

This operation does not accept a request body.

### 3.2.1.2. Response

#### Example 3.5. List 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
    },
    "rate": [
      {
        "limit": [
          {
            "next-available": "2012-11-27T17:22:18Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "POST"
          },
          {
            "next-available": "2012-11-27T17:22:18Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "PUT"
          }
        ]
      }
    ]
  }
}
```

```
{
    "next-available": "2012-11-27T17:22:18Z",
    "remaining": 120,
    "unit": "MINUTE",
    "value": 120,
    "verb": "DELETE"
}
],
"regex": ".*",
"uri": "*"
},
{
"limit": [
{
    "next-available": "2012-11-27T17:22:18Z",
    "remaining": 120,
    "unit": "MINUTE",
    "value": 120,
    "verb": "POST"
}
],
"regex": "^/servers",
"uri": "*/servers"
},
{
"limit": [
{
    "next-available": "2012-11-27T17:22:18Z",
    "remaining": 120,
    "unit": "MINUTE",
    "value": 120,
    "verb": "GET"
}
],
"regex": ".*changes-since.*",
"uri": "*changes-since"
},
{
"limit": [
{
    "next-available": "2012-11-27T17:22:18Z",
    "remaining": 12,
    "unit": "MINUTE",
    "value": 12,
    "verb": "GET"
}
],
"regex": "^/os-fping",
"uri": "*/os-fping"
}
]
}
}
```

### Example 3.6. List 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>
```

```

<rate regex=".*" uri="*">
    <limit next-available="2012-11-27T17:22:18Z" unit="MINUTE" verb="POST"
remaining="120" value="120"/>
    <limit next-available="2012-11-27T17:22:18Z" unit="MINUTE" verb="PUT"
remaining="120" value="120"/>
    <limit next-available="2012-11-27T17:22:18Z" unit="MINUTE" verb="DELETE"
remaining="120" value="120"/>
</rate>
<rate regex="^/servers" uri="*/servers">
    <limit next-available="2012-11-27T17:22:18Z" unit="MINUTE" verb="POST"
remaining="120" value="120"/>
</rate>
<rate regex=".*changes-since.*" uri="*changes-since*>
    <limit next-available="2012-11-27T17:22:18Z" unit="MINUTE" verb="GET"
remaining="120" value="120"/>
</rate>
<rate regex="^/os-fping" uri="*/os-fping">
    <limit next-available="2012-11-27T17:22:18Z" unit="MINUTE" verb="GET"
remaining="12" value="12"/>
</rate>
</rates>
<absolute>
    <limit name="maxServerMeta" value="128"/>
    <limit name="maxPersonality" value="5"/>
    <limit name="maxImageMeta" value="128"/>
    <limit name="maxPersonalitySize" value="10240"/>
    <limit name="maxSecurityGroupRules" value="20"/>
    <limit name="maxTotalKeypairs" value="100"/>
    <limit name="maxSecurityGroups" value="10"/>
    <limit name="maxTotalCores" value="20"/>
    <limit name="maxTotalFloatingIps" value="10"/>
    <limit name="maxTotalInstances" value="10"/>
    <limit name="maxTotalRAMSize" value="51200"/>
</absolute>
</limits>

```

### 3.3. Extensions

List available Compute API v2 extensions and show details for a specified extension.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/extensions	Lists available extensions.
<b>GET</b>	/v2/{tenant_id}/extensions/{alias}	Gets details about the specified extension.

### 3.3.1. List extensions

Method	URI	Description
GET	/v2/{tenant_id}/extensions	Lists available extensions.

**Normal response codes:** 200, 203

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

#### 3.3.1.1. Request

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.

#### 3.3.1.2. Response

##### Example 3.7. List extensions: JSON response

```
{
    "extensions": [
        {
            "alias": "NMN",
            "description": "Multiple network support.",
            "links": [],
            "name": "Multinic",
            "namespace": "http://docs.openstack.org/compute/ext/multinic/api/v1.1",
            "updated": "2011-06-09T00:00:00+00:00"
        },
        {
            "alias": "OS-DCF",
            "description": "Disk Management Extension.",
            "links": [],
            "name": "DiskConfig",
            "namespace": "http://docs.openstack.org/compute/ext/disk_config/api/v1.1",
            "updated": "2011-09-27T00:00:00+00:00"
        },
        {
            "alias": "OS-EXT-AZ",
            "description": "Extended Server Attributes support.",
            "links": [],
            "name": "ExtendedAvailabilityZone",
            "namespace": "http://docs.openstack.org/compute/ext/extended_availability_zone/api/v2",
            "updated": "2013-01-30T00:00:00+00:00"
        },
        {
            "alias": "OS-EXT-IMG-SIZE",
            "description": "Adds image size to image listings.",
            "links": [],
            "name": "ImageSize",
            "namespace": "http://docs.openstack.org/compute/ext/image_size/api/v2",
            "updated": "2013-01-30T00:00:00+00:00"
        }
    ]
}
```

```
        "namespace": "http://docs.openstack.org/compute/ext/image_size/
api/v1.1",
        "updated": "2013-02-19T00:00:00+00:00"
    },
    {
        "alias": "OS-EXT-IPS",
        "description": "Adds type parameter to the ip list.",
        "links": [],
        "name": "ExtendedIps",
        "namespace": "http://docs.openstack.org/compute/ext/extended_ips/
api/v1.1",
        "updated": "2013-01-06T00:00:00+00:00"
    },
    {
        "alias": "OS-EXT-IPS-MAC",
        "description": "Adds mac address parameter to the ip list.",
        "links": [],
        "name": "ExtendedIpsMac",
        "namespace": "http://docs.openstack.org/compute/ext/
extended_ips_mac/api/v1.1",
        "updated": "2013-03-07T00:00:00+00:00"
    },
    {
        "alias": "OS-EXT-SRV-ATTR",
        "description": "Extended Server Attributes support.",
        "links": [],
        "name": "ExtendedServerAttributes",
        "namespace": "http://docs.openstack.org/compute/ext/
extended_status/api/v1.1",
        "updated": "2011-11-03T00:00:00+00:00"
    },
    {
        "alias": "OS-EXT-STS",
        "description": "Extended Status support.",
        "links": [],
        "name": "ExtendedStatus",
        "namespace": "http://docs.openstack.org/compute/ext/
extended_status/api/v1.1",
        "updated": "2011-11-03T00:00:00+00:00"
    },
    {
        "alias": "OS-EXT-VIF-NET",
        "description": "Adds network id parameter to the virtual interface
list.",
        "links": [],
        "name": "ExtendedVIFNet",
        "namespace": "http://docs.openstack.org/compute/ext/extended-
virtual-interfaces-net/api/v1.1",
        "updated": "2013-03-07T00:00:00+00:00"
    },
    {
        "alias": "OS-FLV-DISABLED",
        "description": "Support to show the disabled status of a flavor.",
        "links": [],
        "name": "FlavorDisabled",
        "namespace": "http://docs.openstack.org/compute/ext/
flavor_disabled/api/v1.1",
        "updated": "2012-08-29T00:00:00+00:00"
    },
    {
```

```
        "alias": "OS-FLV-EXT-DATA",
        "description": "Provide additional data for flavors.",
        "links": [],
        "name": "FlavorExtraData",
        "namespace": "http://docs.openstack.org/compute/ext/
flavor_extra_data/api/v1.1",
        "updated": "2011-09-14T00:00:00+00:00"
    },
    {
        "alias": "OS-SCH-HNT",
        "description": "Pass arbitrary key/value pairs to the scheduler.",
        "links": [],
        "name": "SchedulerHints",
        "namespace": "http://docs.openstack.org/compute/ext/scheduler-
hints/api/v2",
        "updated": "2011-07-19T00:00:00+00:00"
    },
    {
        "alias": "OS-SRV-USG",
        "description": "Adds launched_at and terminated_at on Servers.",
        "links": [],
        "name": "ServerUsage",
        "namespace": "http://docs.openstack.org/compute/ext/server_usage/
api/v1.1",
        "updated": "2013-04-29T00:00:00+00:00"
    },
    {
        "alias": "os-admin-actions",
        "description": "Enable admin-only server actions\n\nActions
include: pause, unpause, suspend, resume, migrate,\n            resetNetwork,
            injectNetworkInfo, lock, unlock, createBackup\n        ",
        "links": [],
        "name": "AdminActions",
        "namespace": "http://docs.openstack.org/compute/ext/admin-actions/
api/v1.1",
        "updated": "2011-09-20T00:00:00+00:00"
    },
    {
        "alias": "os-agents",
        "description": "Agents support.",
        "links": [],
        "name": "Agents",
        "namespace": "http://docs.openstack.org/compute/ext/agents/api/
v2",
        "updated": "2012-10-28T00:00:00-00:00"
    },
    {
        "alias": "os-aggregates",
        "description": "Admin-only aggregate administration.",
        "links": [],
        "name": "Aggregates",
        "namespace": "http://docs.openstack.org/compute/ext/aggregates/
api/v1.1",
        "updated": "2012-01-12T00:00:00+00:00"
    },
    {
        "alias": "os-assisted-volume-snapshots",
        "description": "Assisted volume snapshots.",
        "links": [],
        "name": "AssistedVolumeSnapshots",
        "namespace": "http://docs.openstack.org/compute/ext/
assistedvolumesnapshots/api/v1.1",
        "updated": "2012-01-12T00:00:00+00:00"
    }
]
```

```
        "namespace": "http://docs.openstack.org/compute/ext/assisted-
volume-snapshots/api/v2",
        "updated": "2013-08-29T00:00:00-00:00"
    },
    {
        "alias": "os-attach-interfaces",
        "description": "Attach interface support.",
        "links": [],
        "name": "AttachInterfaces",
        "namespace": "http://docs.openstack.org/compute/ext/interfaces/
api/v1.1",
        "updated": "2012-07-22T00:00:00+00:00"
    },
    {
        "alias": "os-availability-zone",
        "description": "1. Add availability_zone to the Create Server v1.1
API.\n          2. Add availability zones describing.\n          ",
        "links": [],
        "name": "AvailabilityZone",
        "namespace": "http://docs.openstack.org/compute/ext/
availabilityzone/api/v1.1",
        "updated": "2012-12-21T00:00:00+00:00"
    },
    {
        "alias": "os-baremetal-ext-status",
        "description": "Add extended status in Baremetal Nodes v2 API.",
        "links": [],
        "name": "BareMetalExtStatus",
        "namespace": "http://docs.openstack.org/compute/ext/
baremetal_ext_status/api/v2",
        "updated": "2013-08-27T00:00:00+00:00"
    },
    {
        "alias": "os-baremetal-nodes",
        "description": "Admin-only bare-metal node administration.",
        "links": [],
        "name": "BareMetalNodes",
        "namespace": "http://docs.openstack.org/compute/ext/
baremetal_nodes/api/v2",
        "updated": "2013-01-04T00:00:00+00:00"
    },
    {
        "alias": "os-block-device-mapping-v2-boot",
        "description": "Allow boot with the new BDM data format.",
        "links": [],
        "name": "BlockDeviceMappingV2Boot",
        "namespace": "http://docs.openstack.org/compute/ext/
block_device_mapping_v2_boot/api/v2",
        "updated": "2013-07-08T00:00:00+00:00"
    },
    {
        "alias": "os-cell-capacities",
        "description": "Adding functionality to get cell capacities.",
        "links": [],
        "name": "CellCapacities",
        "namespace": "http://docs.openstack.org/compute/ext/
cell_capacities/api/v1.1",
        "updated": "2013-05-27T00:00:00+00:00"
    },
    {
```

```
        "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/cells/api/v1.
1",
        "updated": "2013-05-14T00:00:00+00:00"
    },
    {
        "alias": "os-certificates",
        "description": "Certificates support.",
        "links": [],
        "name": "Certificates",
        "namespace": "http://docs.openstack.org/compute/ext/certificates/
api/v1.1",
        "updated": "2012-01-19T00:00:00+00:00"
    },
    {
        "alias": "os-cloudpipe",
        "description": "Adds actions to create cloudpipe instances.\n\n      When running with the Vlan network mode, you need a mechanism to route\
n      from the public Internet to your vlans. This mechanism is known as a\
n      cloudpipe.\n\n      At the time of creating this class, only OpenVPN is
supported. Support for\n      a SSH Bastion host is forthcoming.\n      ",
        "links": [],
        "name": "Cloudpipe",
        "namespace": "http://docs.openstack.org/compute/ext/cloudpipe/api/
v1.1",
        "updated": "2011-12-16T00:00:00+00:00"
    },
    {
        "alias": "os-cloudpipe-update",
        "description": "Adds the ability to set the vpn ip/port for
cloudpipe instances.",
        "links": [],
        "name": "CloudpipeUpdate",
        "namespace": "http://docs.openstack.org/compute/ext/cloudpipe-
update/api/v2",
        "updated": "2012-11-14T00:00:00+00:00"
    },
    {
        "alias": "os-config-drive",
        "description": "Config Drive Extension.",
        "links": [],
        "name": "ConfigDrive",
        "namespace": "http://docs.openstack.org/compute/ext/config_drive/
api/v1.1",
        "updated": "2012-07-16T00:00:00+00:00"
    },
    {
        "alias": "os-console-output",
        "description": "Console log output support, with tailing ability.
",
        "links": [],
        "name": "ConsoleOutput",
        "namespace": "http://docs.openstack.org/compute/ext/os-console-
output/api/v2",
        "updated": "2011-12-08T00:00:00+00:00"
```

```
        },
        {
            "alias": "os-consoles",
            "description": "Interactive Console support.",
            "links": [],
            "name": "Consoles",
            "namespace": "http://docs.openstack.org/compute/ext/os-consoles/
api/v2",
            "updated": "2011-12-23T00:00:00+00:00"
        },
        {
            "alias": "os-create-server-ext",
            "description": "Extended support to the Create Server v1.1 API.",
            "links": [],
            "name": "Createserverext",
            "namespace": "http://docs.openstack.org/compute/ext/
createserverext/api/v1.1",
            "updated": "2011-07-19T00:00:00+00:00"
        },
        {
            "alias": "os-deferred-delete",
            "description": "Instance deferred delete.",
            "links": [],
            "name": "DeferredDelete",
            "namespace": "http://docs.openstack.org/compute/ext/deferred-
delete/api/v1.1",
            "updated": "2011-09-01T00:00:00+00:00"
        },
        {
            "alias": "os-evacuate",
            "description": "Enables server evacuation.",
            "links": [],
            "name": "Evacuate",
            "namespace": "http://docs.openstack.org/compute/ext/evacuate/api/
v2",
            "updated": "2013-01-06T00:00:00+00:00"
        },
        {
            "alias": "os-extended-floating-ips",
            "description": "Adds optional fixed_address to the add floating IP
command.",
            "links": [],
            "name": "ExtendedFloatingIps",
            "namespace": "http://docs.openstack.org/compute/ext/
extended_floating_ips/api/v2",
            "updated": "2013-04-19T00:00:00+00:00"
        },
        {
            "alias": "os-extended-quotas",
            "description": "Adds ability for admins to delete quota\n and
optionally force the update Quota command.\n      ",
            "links": [],
            "name": "ExtendedQuotas",
            "namespace": "http://docs.openstack.org/compute/ext/
extended_quotas/api/v1.1",
            "updated": "2013-06-09T00:00:00+00:00"
        },
        {
            "alias": "os-extended-hypervisors",
            "description": "Extended hypervisors support."
        }
    ]
}
```

```
        "links": [],
        "name": "ExtendedHypervisors",
        "namespace": "http://docs.openstack.org/compute/ext/
extended_hypervisors/api/v1.1",
        "updated": "2013-10-21T00:00:00-00:00"
    },
    {
        "alias": "os-extended-services",
        "description": "Extended services support.",
        "links": [],
        "name": "ExtendedServices",
        "namespace": "http://docs.openstack.org/compute/ext/
extended_services/api/v2",
        "updated": "2013-05-17T00:00:00-00:00"
    },
    {
        "alias": "os-extended-volumes",
        "description": "Extended Volumes support.",
        "links": [],
        "name": "ExtendedVolumes",
        "namespace": "http://docs.openstack.org/compute/ext/
extended_volumes/api/v1.1",
        "updated": "2013-06-07T00:00:00+00:00"
    },
    {
        "alias": "os-fixed-ips",
        "description": "Fixed IPs support.",
        "links": [],
        "name": "FixedIPs",
        "namespace": "http://docs.openstack.org/compute/ext/fixed_ips/api/
v2",
        "updated": "2012-10-18T13:25:27-06:00"
    },
    {
        "alias": "os-flavor-access",
        "description": "Flavor access support.",
        "links": [],
        "name": "FlavorAccess",
        "namespace": "http://docs.openstack.org/compute/ext/flavor_access/
api/v2",
        "updated": "2012-08-01T00:00:00+00:00"
    },
    {
        "alias": "os-flavor-extra-specs",
        "description": "Instance type (flavor) extra specs.",
        "links": [],
        "name": "FlavorExtraSpecs",
        "namespace": "http://docs.openstack.org/compute/ext/
flavor_extra_specs/api/v1.1",
        "updated": "2011-06-23T00:00:00+00:00"
    },
    {
        "alias": "os-flavor-manage",
        "description": "\n          Flavor create/delete API support\n        ",
        "links": [],
        "name": "FlavorManage",
        "namespace": "http://docs.openstack.org/compute/ext/flavor_manage/
api/v1.1",
        "updated": "2012-01-19T00:00:00+00:00"
    },
}
```

```
{  
    "alias": "os-flavor-rxtx",  
    "description": "Support to show the rxtx status of a flavor.",  
    "links": [],  
    "name": "FlavorRxtx",  
    "namespace": "http://docs.openstack.org/compute/ext/flavor_rxtx/  
api/v1.1",  
    "updated": "2012-08-29T00:00:00+00:00"  
},  
{  
    "alias": "os-flavor-swap",  
    "description": "Support to show the swap status of a flavor.",  
    "links": [],  
    "name": "FlavorSwap",  
    "namespace": "http://docs.openstack.org/compute/ext/flavor_swap/  
api/v1.1",  
    "updated": "2012-08-29T00:00:00+00:00"  
},  
{  
    "alias": "os-floating-ip-dns",  
    "description": "Floating IP DNS support.",  
    "links": [],  
    "name": "FloatingIpDns",  
    "namespace": "http://docs.openstack.org/ext/floating_ip_dns/api/  
v1.1",  
    "updated": "2011-12-23T00:00:00+00:00"  
},  
{  
    "alias": "os-floating-ip-pools",  
    "description": "Floating IPs support.",  
    "links": [],  
    "name": "FloatingIpPools",  
    "namespace": "http://docs.openstack.org/compute/ext/  
floating_ip_pools/api/v1.1",  
    "updated": "2012-01-04T00:00:00+00:00"  
},  
{  
    "alias": "os-floating-ips",  
    "description": "Floating IPs support.",  
    "links": [],  
    "name": "FloatingIps",  
    "namespace": "http://docs.openstack.org/compute/ext/floating_ips/  
api/v1.1",  
    "updated": "2011-06-16T00:00:00+00:00"  
},  
{  
    "alias": "os-floating-ips-bulk",  
    "description": "Bulk handling of Floating IPs.",  
    "links": [],  
    "name": "FloatingIpsBulk",  
    "namespace": "http://docs.openstack.org/compute/ext/  
floating_ips_bulk/api/v2",  
    "updated": "2012-10-29T13:25:27-06:00"  
},  
{  
    "alias": "os-fping",  
    "description": "Fping Management Extension.",  
    "links": [],  
    "name": "Fping",  
    "namespace": "http://docs.openstack.org/compute/ext/fping/api/v1.1",  
    "updated": "2012-08-29T00:00:00+00:00"  
}
```

```
        "namespace": "http://docs.openstack.org/compute/ext/fping/api/v1.
1",
        "updated": "2012-07-06T00:00:00+00:00"
    },
    {
        "alias": "os-hide-server-addresses",
        "description": "Support hiding server addresses in certain states.
",
        "links": [],
        "name": "HideServerAddresses",
        "namespace": "http://docs.openstack.org/compute/ext/
hide_server_addresses/api/v1.1",
        "updated": "2012-12-11T00:00:00+00:00"
    },
    {
        "alias": "os-hosts",
        "description": "Admin-only host administration.",
        "links": [],
        "name": "Hosts",
        "namespace": "http://docs.openstack.org/compute/ext/hosts/api/v1.
1",
        "updated": "2011-06-29T00:00:00+00:00"
    },
    {
        "alias": "os-hypervisors",
        "description": "Admin-only hypervisor administration.",
        "links": [],
        "name": "Hypervisors",
        "namespace": "http://docs.openstack.org/compute/ext/hypervisors/
api/v1.1",
        "updated": "2012-06-21T00:00:00+00:00"
    },
    {
        "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/instance-
actions/api/v1.1",
        "updated": "2013-02-08T00:00:00+00:00"
    },
    {
        "alias": "os-server-external-events",
        "description": "Server External Event Triggers.",
        "links": [],
        "name": "ServerExternalEvents",
        "namespace": "http://docs.openstack.org/compute/ext/server-
external-events/api/v2",
        "updated": "2014-02-18T00:00:00-00:00"
    },
    {
        "alias": "os-server-groups",
        "description": "Server group support.",
        "links": [],
        "name": "ServerGroups",
        "namespace": "http://docs.openstack.org/compute/ext/servergroups/
api/v2",
        "updated": "2013-06-01T00:00:00+00:00"
    },
}
```

```
{  
    "alias": "os-instance_usage_audit_log",  
    "description": "Admin-only Task Log Monitoring.",  
    "links": [],  
    "name": "OSInstanceUsageAuditLog",  
    "namespace": "http://docs.openstack.org/ext/services/api/v1.1",  
    "updated": "2012-07-06T01:00:00+00:00"  
},  
{  
    "alias": "os-keypairs",  
    "description": "Keypair Support.",  
    "links": [],  
    "name": "Keypairs",  
    "namespace": "http://docs.openstack.org/compute/ext/keypairs/api/  
v1.1",  
    "updated": "2011-08-08T00:00:00+00:00"  
},  
{  
    "alias": "os-migrations",  
    "description": "Provide data on migrations.",  
    "links": [],  
    "name": "Migrations",  
    "namespace": "http://docs.openstack.org/compute/ext/migrations/  
api/v2.0",  
    "updated": "2013-05-30T00:00:00+00:00"  
},  
{  
    "alias": "os-multiple-create",  
    "description": "Allow multiple create in the Create Server v1.1  
API.",  
    "links": [],  
    "name": "MultipleCreate",  
    "namespace": "http://docs.openstack.org/compute/ext/  
multiplecreate/api/v1.1",  
    "updated": "2012-08-07T00:00:00+00:00"  
},  
{  
    "alias": "os-networks",  
    "description": "Admin-only Network Management Extension.",  
    "links": [],  
    "name": "Networks",  
    "namespace": "http://docs.openstack.org/compute/ext/os-networks/  
api/v1.1",  
    "updated": "2011-12-23T00:00:00+00:00"  
},  
{  
    "alias": "os-networks-associate",  
    "description": "Network association support.",  
    "links": [],  
    "name": "NetworkAssociationSupport",  
    "namespace": "http://docs.openstack.org/compute/ext/  
networks_associate/api/v2",  
    "updated": "2012-11-19T00:00:00+00:00"  
},  
{  
    "alias": "os-preserve-ephemeral-rebuild",  
    "description": "Allow preservation of the ephemeral partition on  
rebuild.",  
    "links": [],  
    "name": "PreserveEphemeralOnRebuild",  
    "namespace": "http://docs.openstack.org/compute/ext/  
preserve_ephemeral_rebuild/api/v1.1",  
    "updated": "2012-11-19T00:00:00+00:00"  
}
```

```
        "namespace": "http://docs.openstack.org/compute/ext/preserve_ephemeral_rebuild/api/v2",
        "updated": "2013-12-17T00:00:00+00:00"
    },
    {
        "alias": "os-quota-sets",
        "description": "Quotas management support.",
        "links": [],
        "name": "Quotas",
        "namespace": "http://docs.openstack.org/compute/ext/quotas-sets/api/v1.1",
        "updated": "2011-08-08T00:00:00+00:00"
    },
    {
        "alias": "os-rescue",
        "description": "Instance rescue mode.",
        "links": [],
        "name": "Rescue",
        "namespace": "http://docs.openstack.org/compute/ext/rescue/api/v1.1",
        "updated": "2011-08-18T00:00:00+00:00"
    },
    {
        "alias": "os-extended-rescue-with-image",
        "description": "Rescue instance with the image specified.",
        "links": [],
        "name": "ExtendedRescueWithImage",
        "namespace": "http://docs.openstack.org/compute/ext/extended_rescue_with_image/api/v2",
        "updated": "2014-03-05T00:00:00+00:00"
    },
    {
        "alias": "os-security-group-default-rules",
        "description": "Default rules for security group support.",
        "links": [],
        "name": "SecurityGroupDefaultRules",
        "namespace": "http://docs.openstack.org/compute/ext/securitygroupdefaultrules/api/v1.1",
        "updated": "2013-02-05T00:00:00+00:00"
    },
    {
        "alias": "os-security-groups",
        "description": "Security group support.",
        "links": [],
        "name": "SecurityGroups",
        "namespace": "http://docs.openstack.org/compute/ext/securitygroups/api/v1.1",
        "updated": "2013-05-28T00:00:00+00:00"
    },
    {
        "alias": "os-server-diagnostics",
        "description": "Allow Admins to view server diagnostics through server action.",
        "links": [],
        "name": "ServerDiagnostics",
        "namespace": "http://docs.openstack.org/compute/ext/server-diagnostics/api/v1.1",
        "updated": "2011-12-21T00:00:00+00:00"
    },
    {

```

```
        "alias": "os-server-password",
        "description": "Server password support.",
        "links": [],
        "name": "ServerPassword",
        "namespace": "http://docs.openstack.org/compute/ext/server-
password/api/v2",
        "updated": "2012-11-29T00:00:00+00:00"
    },
    {
        "alias": "os-server-start-stop",
        "description": "Start/Stop instance compute API support.",
        "links": [],
        "name": "ServerStartStop",
        "namespace": "http://docs.openstack.org/compute/ext/servers/api/
v1.1",
        "updated": "2012-01-23T00:00:00+00:00"
    },
    {
        "alias": "os-services",
        "description": "Services support.",
        "links": [],
        "name": "Services",
        "namespace": "http://docs.openstack.org/compute/ext/services/api/
v2",
        "updated": "2012-10-28T00:00:00-00:00"
    },
    {
        "alias": "os-shelve",
        "description": "Instance shelve mode.",
        "links": [],
        "name": "Shelve",
        "namespace": "http://docs.openstack.org/compute/ext/shelve/api/v1.
1",
        "updated": "2013-04-06T00:00:00+00:00"
    },
    {
        "alias": "os-simple-tenant-usage",
        "description": "Simple tenant usage extension.",
        "links": [],
        "name": "SimpleTenantUsage",
        "namespace": "http://docs.openstack.org/compute/ext/os-simple-
tenant-usage/api/v1.1",
        "updated": "2011-08-19T00:00:00+00:00"
    },
    {
        "alias": "os-tenant-networks",
        "description": "Tenant-based Network Management Extension.",
        "links": [],
        "name": "OSTenantNetworks",
        "namespace": "http://docs.openstack.org/compute/ext/os-tenant-
networks/api/v2",
        "updated": "2012-03-07T09:46:43-05:00"
    },
    {
        "alias": "os-used-limits",
        "description": "Provide data on limited resources that are being
used.",
        "links": [],
        "name": "UsedLimits",
        "namespace": "http://docs.openstack.org/compute/ext/os-used-limits/api/v1.1"
    }
]
```

```
        "namespace": "http://docs.openstack.org/compute/ext/used_limits/
api/v1.1",
        "updated": "2012-07-13T00:00:00+00:00"
    },
    {
        "alias": "os-used-limits-for-admin",
        "description": "Provide data to admin on limited resources used by
other tenants.",
        "links": [],
        "name": "UsedLimitsForAdmin",
        "namespace": "http://docs.openstack.org/compute/ext/
used_limits_for_admin/api/v1.1",
        "updated": "2013-05-02T00:00:00+00:00"
    },
    {
        "alias": "os-user-data",
        "description": "Add user_data to the Create Server v1.1 API.",
        "links": [],
        "name": "UserData",
        "namespace": "http://docs.openstack.org/compute/ext/userdata/api/
v1.1",
        "updated": "2012-08-07T00:00:00+00:00"
    },
    {
        "alias": "os-user-quotas",
        "description": "Project user quota support.",
        "links": [],
        "name": "UserQuotas",
        "namespace": "http://docs.openstack.org/compute/ext/user_quotas/
api/v1.1",
        "updated": "2013-07-18T00:00:00+00:00"
    },
    {
        "alias": "os-virtual-interfaces",
        "description": "Virtual interface support.",
        "links": [],
        "name": "VirtualInterfaces",
        "namespace": "http://docs.openstack.org/compute/ext/
virtual_interfaces/api/v1.1",
        "updated": "2011-08-17T00:00:00+00:00"
    },
    {
        "alias": "os-volume-attachment-update",
        "description": "Support for updating a volume attachment.",
        "links": [],
        "name": "VolumeAttachmentUpdate",
        "namespace": "http://docs.openstack.org/compute/ext/os-volume-
attachment-update/api/v2",
        "updated": "2013-06-20T00:00:00-00:00"
    },
    {
        "alias": "os-volumes",
        "description": "Volumes support.",
        "links": [],
        "name": "Volumes",
        "namespace": "http://docs.openstack.org/compute/ext/volumes/api/
v1.1",
        "updated": "2011-03-25T00:00:00+00:00"
    },
    {
```

```

        "alias": "os-extended-services-delete",
        "description": "Services deletion support.",
        "links": [],
        "name": "ExtendedServicesDelete",
        "namespace": "http://docs.openstack.org/compute/ext/
extended_services_delete/api/v2",
        "updated": "2013-12-10T00:00:00+00:00"
    },
    {
        "alias": "os-console-auth-tokens",
        "description": "Console token authentication support.",
        "links": [],
        "name": "ConsoleAuthTokens",
        "namespace": "http://docs.openstack.org/compute/ext/consoles-auth-
tokens/api/v2",
        "updated": "2013-08-13T00:00:00+00:00"
    }
]
}

```

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

Name	Type	Description
next	AnyURI <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI <i>(Optional)</i>	Moves to the previous metadata item.

### Example 3.8. List 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="NMN" updated="2011-06-09T00:00:00+00:00"
               namespace="http://docs.openstack.org/compute/ext/multinic/api/v1.1"
               name="Multinic">
        <description>Multiple network support.</description>
    </extension>
    <extension alias="OS-DCF" updated="2011-09-27T00:00:00+00:00"
               namespace="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
               name="DiskConfig">
        <description>Disk Management Extension.</description>
    </extension>
    <extension alias="OS-EXT-AZ" updated="2013-01-30T00:00:00+00:00"
               namespace="http://docs.openstack.org/compute/ext/
extended_availability_zone/api/v2"
               name="ExtendedAvailabilityZone">
        <description>Extended Availability Zone support.</description>
    </extension>
    <extension alias="OS-EXT-IMG-SIZE"
               updated="2013-02-19T00:00:00+00:00"
               namespace="http://docs.openstack.org/compute/ext/image_size/api/v1.1"
               name="ImageSize">
        <description>Adds image size to image listings.</description>
    </extension>
    <extension alias="OS-EXT-IPS" updated="2013-01-06T00:00:00+00:00"
               namespace="http://docs.openstack.org/compute/ext/extended_ips/api/v1.1"
               name="ExtendedIps">

```

```
<description>Adds type parameter to the ip list.</description>
</extension>
<extension alias="OS-EXT-IPS-MAC"
    updated="2013-03-07T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_ips_mac/api/v1.
1"
    name="ExtendedIpsMac">
    <description>Adds mac address parameter to the ip
        list.</description>
</extension>
<extension alias="OS-EXT-SRV-ATTR"
    updated="2011-11-03T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_status/api/v1.1"
    name="ExtendedServerAttributes">
    <description>Extended Server Attributes support.</description>
</extension>
<extension alias="OS-EXT-STS" updated="2011-11-03T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_status/api/v1.1"
    name="ExtendedStatus">
    <description>Extended Status support.</description>
</extension>
<extension alias="OS-EXT-VIF-NET"
    updated="2013-03-07T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/extended-virtual-
interfaces-net/api/v1.1"
    name="ExtendedVIFNet">
    <description>Adds network id parameter to the virtual interface
        list.</description>
</extension>
<extension alias="OS-FLV-DISABLED"
    updated="2012-08-29T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/flavor_disabled/api/v1.1"
    name="FlavorDisabled">
    <description>Support to show the disabled status of a
        flavor.</description>
</extension>
<extension alias="OS-FLV-EXT-DATA"
    updated="2011-09-14T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/flavor_extra_data/api/v1.
1"
    name="FlavorExtraData">
    <description>Provide additional data for flavors.</description>
</extension>
<extension alias="OS-SCH-HNT" updated="2011-07-19T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/scheduler-hints/api/v2"
    name="SchedulerHints">
    <description>Pass arbitrary key/value pairs to the
        scheduler.</description>
</extension>
<extension alias="OS-SRV-USG" updated="2013-04-29T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/server_usage/api/v1.1"
    name="ServerUsage">
    <description>Adds launched_at and terminated_at on
        Servers.</description>
</extension>
<extension alias="os-admin-actions"
    updated="2011-09-20T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/admin-actions/api/v1.1"
    name="AdminActions">
    <description>Enable admin-only server actions Actions include:
```

```
        pause, unpause, suspend, resume, migrate, resetNetwork,
        injectNetworkInfo, lock, unlock, createBackup </description>
</extension>
<extension alias="os-agents" updated="2012-10-28T00:00:00-00:00"
    namespace="http://docs.openstack.org/compute/ext/agents/api/v2"
    name="Agents">
    <description>Agents support.</description>
</extension>
<extension alias="os-aggregates" updated="2012-01-12T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/aggregates/api/v1.1"
    name="Aggregates">
    <description>Admin-only aggregate administration.</description>
</extension>
<extension alias="os-assisted-volume-snapshots"
    updated="2013-08-29T00:00:00-00:00"
    namespace="http://docs.openstack.org/compute/ext/assisted-volume-
snapshots/api/v2"
    name="AssistedVolumeSnapshots">
    <description>Assisted volume snapshots.</description>
</extension>
<extension alias="os-attach-interfaces"
    updated="2012-07-22T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/interfaces/api/v1.1"
    name="AttachInterfaces">
    <description>Attach interface support.</description>
</extension>
<extension alias="os-availability-zone"
    updated="2012-12-21T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/availabilityzone/api/v1.
1"
    name="AvailabilityZone">
    <description>1. Add availability_zone to the Create Server v1.1
        API. 2. Add availability zones describing. </description>
</extension>
<extension alias="os-baremetal-ext-status"
    updated="2013-08-27T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/baremetal_ext_status/api/
v2"
    name="BareMetalExtStatus">
    <description>Add extended status in Baremetal Nodes v2
        API.</description>
</extension>
<extension alias="os-baremetal-nodes"
    updated="2013-01-04T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/baremetal_nodes/api/v2"
    name="BareMetalNodes">
    <description>Admin-only bare-metal node
        administration.</description>
</extension>
<extension alias="os-block-device-mapping-v2-boot"
    updated="2013-07-08T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/
block_device_mapping_v2_boot/api/v2"
    name="BlockDeviceMappingV2Boot">
    <description>Allow boot with the new BDM data
        format.</description>
</extension>
<extension alias="os-cell-capacities"
    updated="2013-05-27T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/cell_capacities/api/v1.1"
```

```
name="CellCapacities">
<description>Adding functionality to get cell
    capacities.</description>
</extension>
<extension alias="os-cells" updated="2013-05-14T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/cells/api/v1.1"
    name="Cells">
    <description>Enables cells-related functionality such as adding
        neighbor cells, listing neighbor cells, and getting the
        capabilities of the local cell. </description>
</extension>
<extension alias="os-certificates"
    updated="2012-01-19T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/certificates/api/v1.1"
    name="Certificates">
    <description>Certificates support.</description>
</extension>
<extension alias="os-cloudpipe" updated="2011-12-16T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/cloudpipe/api/v1.1"
    name="Cloudpipe">
    <description>Adds actions to create cloudpipe instances. When
        running with the Vlan network mode, you need a mechanism to
        route from the public Internet to your vlans. This mechanism is
        known as a cloudpipe. At the time of creating this class, only
        OpenVPN is supported. Support for a SSH Bastion host is
        forthcoming. </description>
</extension>
<extension alias="os-cloudpipe-update"
    updated="2012-11-14T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/cloudpipe-update/api/v2"
    name="CloudpipeUpdate">
    <description>Adds the ability to set the vpn ip/port for cloudpipe
        instances.</description>
</extension>
<extension alias="os-config-drive"
    updated="2012-07-16T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/config_drive/api/v1.1"
    name="ConfigDrive">
    <description>Config Drive Extension.</description>
</extension>
<extension alias="os-console-auth-tokens"
    updated="2013-08-13T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/consoles-auth-tokens/api/
v2"
    name="ConsoleAuthTokens">
    <description>Console token authentication support.</description>
</extension>
<extension alias="os-console-output"
    updated="2011-12-08T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/os-console-output/api/v2"
    name="ConsoleOutput">
    <description>Console log output support, with tailing
        ability.</description>
</extension>
<extension alias="os-consoles" updated="2011-12-23T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/os-consoles/api/v2"
    name="Consoles">
    <description>Interactive Console support.</description>
</extension>
<extension alias="os-create-server-ext"
```

```
updated="2011-07-19T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/createserverext/api/v1.1"
name="Createserverext">
<description>Extended support to the Create Server v1.1
    API.</description>
</extension>
<extension alias="os-deferred-delete"
    updated="2011-09-01T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/deferred-delete/api/v1.1"
    name="DeferredDelete">
    <description>Instance deferred delete.</description>
</extension>
<extension alias="os-evacuate" updated="2013-01-06T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/evacuate/api/v2"
    name="Evacuate">
    <description>Enables server evacuation.</description>
</extension>
<extension alias="os-extended-floating-ips"
    updated="2013-04-19T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_floating_ips/
api/v2"
    name="ExtendedFloatingIps">
    <description>Adds optional fixed_address to the add floating IP
        command.</description>
</extension>
<extension alias="os-extended-hypervisors"
    updated="2014-01-04T00:00:00-00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_hypervisors/api/
v1.1"
    name="ExtendedHypervisors">
    <description>Extended hypervisors support.</description>
</extension>
<extension alias="os-extended-quotas"
    updated="2013-06-09T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_quotas/api/v1.1"
    name="ExtendedQuotas">
    <description>Adds ability for admins to delete quota and
        optionally force the update Quota command. </description>
</extension>
<extension alias="os-extended-services"
    updated="2013-05-17T00:00:00-00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_services/api/v2"
    name="ExtendedServices">
    <description>Extended services support.</description>
</extension>
<extension alias="os-extended-services-delete"
    updated="2013-12-10T00:00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_services_delete/
api/v2"
    name="ExtendedServicesDelete">
    <description>Extended services deletion support.</description>
</extension>
<extension alias="os-extended-volumes"
    updated="2013-06-07T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_volumes/api/v1.
1"
    name="ExtendedVolumes">
    <description>Extended Volumes support.</description>
</extension>
<extension alias="os-fixed-ips" updated="2012-10-18T13:25:27-06:00"
```

```
namespace="http://docs.openstack.org/compute/ext/fixed_ips/api/v2"
  name="FixedIPs">
    <description>Fixed IPs support.</description>
</extension>
<extension alias="os-flavor-access"
  updated="2012-08-01T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/flavor_access/api/v2"
  name="FlavorAccess">
    <description>Flavor access support.</description>
</extension>
<extension alias="os-flavor-extra-specs"
  updated="2011-06-23T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/flavor_extra_specs/api/
v1.1"
  name="FlavorExtraSpecs">
    <description>Instance type (flavor) extra specs.</description>
</extension>
<extension alias="os-flavor-manage"
  updated="2012-01-19T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/flavor_manage/api/v1.1"
  name="FlavorManage">
    <description>Flavor create/delete API support.</description>
</extension>
<extension alias="os-flavor-rxtx"
  updated="2012-08-29T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/flavor_rxtx/api/v1.1"
  name="FlavorRxtx">
    <description>Support to show the rxtx status of a
      flavor.</description>
</extension>
<extension alias="os-flavor-swap"
  updated="2012-08-29T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/flavor_swap/api/v1.1"
  name="FlavorSwap">
    <description>Support to show the swap status of a
      flavor.</description>
</extension>
<extension alias="os-floating-ip-dns"
  updated="2011-12-23T00:00:00+00:00"
  namespace="http://docs.openstack.org/ext/floating_ip_dns/api/v1.1"
  name="FloatingIpDns">
    <description>Floating IP DNS support.</description>
</extension>
<extension alias="os-floating-ip-pools"
  updated="2012-01-04T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/floating_ip_pools/api/v1.
1"
  name="FloatingIpPools">
    <description>Floating IPs support.</description>
</extension>
<extension alias="os-floating-ips"
  updated="2011-06-16T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/floating_ips/api/v1.1"
  name="FloatingIps">
    <description>Floating IPs support.</description>
</extension>
<extension alias="os-floating-ips-bulk"
  updated="2012-10-29T13:25:27-06:00"
  namespace="http://docs.openstack.org/compute/ext/floating_ips_bulk/api/v2"
  name="FloatingIpsBulk">
```

```
<description>Bulk handling of Floating IPs.</description>
</extension>
<extension alias="os-fping" updated="2012-07-06T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/fping/api/v1.1"
    name="Fping">
    <description>Fping Management Extension.</description>
</extension>
<extension alias="os-hide-server-addresses"
    updated="2012-12-11T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/hide_server_addresses/
api/v1.1"
    name="HideServerAddresses">
    <description>Support hiding server addresses in certain
        states.</description>
</extension>
<extension alias="os-hosts" updated="2011-06-29T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/hosts/api/v1.1"
    name="Hosts">
    <description>Admin-only host administration.</description>
</extension>
<extension alias="os-hypervisors"
    updated="2012-06-21T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/hypervisors/api/v1.1"
    name="Hypervisors">
    <description>Admin-only hypervisor administration.</description>
</extension>
<extension alias="os-instance-actions"
    updated="2013-02-08T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/instance-actions/api/v1.
1"
    name="InstanceActions">
    <description>View a log of actions and events taken on an
        instance.</description>
</extension>
<extension alias="os-instance_usage_audit_log"
    updated="2012-07-06T01:00:00+00:00"
    namespace="http://docs.openstack.org/ext/services/api/v1.1"
    name="OSInstanceUsageAuditLog">
    <description>Admin-only Task Log Monitoring.</description>
</extension>
<extension alias="os-keypairs" updated="2011-08-08T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/keypairs/api/v1.1"
    name="Keypairs">
    <description>Keypair Support.</description>
</extension>
<extension alias="os-migrations" updated="2013-05-30T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/migrations/api/v2.0"
    name="Migrations">
    <description>Provide data on migrations.</description>
</extension>
<extension alias="os-multiple-create"
    updated="2012-08-07T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/multiplecreate/api/v1.1"
    name="MultipleCreate">
    <description>Allow multiple create in the Create Server v1.1
        API.</description>
</extension>
<extension alias="os-networks" updated="2011-12-23T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/os-networks/api/v1.1"
    name="Networks">
```

```
<description>Admin-only Network Management  
Extension.</description>  
</extension>  
<extension alias="os-networks-associate"  
updated="2012-11-19T00:00:00+00:00"  
namespace="http://docs.openstack.org/compute/ext/networks_associate/api/  
v2"  
name="NetworkAssociationSupport">  
  <description>Network association support.</description>  
</extension>  
<extension alias="os-preserve-ephemeral-rebuild"  
updated="2013-12-17T00:00:00+00:00"  
namespace="http://docs.openstack.org/compute/ext/  
preserve_ephemeral_rebuild/api/v2"  
name="PreserveEphemeralOnRebuild">  
  <description>Allow preservation of the ephemeral partition on  
  rebuild.</description>  
</extension>  
<extension alias="os-quota-sets" updated="2011-08-08T00:00:00+00:00"  
namespace="http://docs.openstack.org/compute/ext/quotas-sets/api/v1.1"  
name="Quotas">  
  <description>Quotas management support.</description>  
</extension>  
<extension alias="os-rescue" updated="2011-08-18T00:00:00+00:00"  
namespace="http://docs.openstack.org/compute/ext/rescue/api/v1.1"  
name="Rescue">  
  <description>Instance rescue mode.</description>  
</extension>  
<extension alias="os-security-group-default-rules"  
updated="2013-02-05T00:00:00+00:00"  
namespace="http://docs.openstack.org/compute/ext/  
securitygroupdefaultrules/api/v1.1"  
name="SecurityGroupDefaultRules">  
  <description>Default rules for security group  
  support.</description>  
</extension>  
<extension alias="os-security-groups"  
updated="2013-05-28T00:00:00+00:00"  
namespace="http://docs.openstack.org/compute/ext/securitygroups/api/v1.1"  
name="SecurityGroups">  
  <description>Security group support.</description>  
</extension>  
<extension alias="os-server-diagnostics"  
updated="2011-12-21T00:00:00+00:00"  
namespace="http://docs.openstack.org/compute/ext/server-diagnostics/api/  
v1.1"  
name="ServerDiagnostics">  
  <description>Allow Admins to view server diagnostics through  
  server action.</description>  
</extension>  
<extension alias="os-server-external-events"  
updated="2014-02-18T00:00:00-00:00"  
namespace="http://docs.openstack.org/compute/ext/server-external-events/  
api/v2"  
name="ServerExternalEvents">  
  <description>Server External Event Triggers.</description>  
</extension>  
<extension alias="os-server-groups"  
updated="2013-06-20T00:00:00+00:00"  
namespace="http://docs.openstack.org/compute/ext/servergroups/api/v2"
```

```
    name="ServerGroups">
      <description>Server group support.</description>
    </extension>
    <extension alias="os-server-password"
      updated="2012-11-29T00:00:00+00:00"
      namespace="http://docs.openstack.org/compute/ext/server-password/api/v2"
      name="ServerPassword">
      <description>Server password support.</description>
    </extension>
    <extension alias="os-server-start-stop"
      updated="2012-01-23T00:00:00+00:00"
      namespace="http://docs.openstack.org/compute/ext/servers/api/v1.1"
      name="ServerStartStop">
      <description>Start/Stop instance compute API
        support.</description>
    </extension>
    <extension alias="os-services" updated="2012-10-28T00:00:00-00:00"
      namespace="http://docs.openstack.org/compute/ext/services/api/v2"
      name="Services">
      <description>Services support.</description>
    </extension>
    <extension alias="os-shelve" updated="2013-04-06T00:00:00+00:00"
      namespace="http://docs.openstack.org/compute/ext/shelve/api/v1.1"
      name="Shelve">
      <description>Instance shelve mode.</description>
    </extension>
    <extension alias="os-simple-tenant-usage"
      updated="2011-08-19T00:00:00+00:00"
      namespace="http://docs.openstack.org/compute/ext/os-simple-tenant-usage/
api/v1.1"
      name="SimpleTenantUsage">
      <description>Simple tenant usage extension.</description>
    </extension>
    <extension alias="os-tenant-networks"
      updated="2012-03-07T09:46:43-05:00"
      namespace="http://docs.openstack.org/compute/ext/os-tenant-networks/api/
v2"
      name="OSTenantNetworks">
      <description>Tenant-based Network Management
        Extension.</description>
    </extension>
    <extension alias="os-used-limits"
      updated="2012-07-13T00:00:00+00:00"
      namespace="http://docs.openstack.org/compute/ext/used_limits/api/v1.1"
      name="UsedLimits">
      <description>Provide data on limited resources that are being
        used.</description>
    </extension>
    <extension alias="os-used-limits-for-admin"
      updated="2013-05-02T00:00:00+00:00"
      namespace="http://docs.openstack.org/compute/ext/used_limits_for_admin/
api/v1.1"
      name="UsedLimitsForAdmin">
      <description>Provide data to admin on limited resources used by
        other tenants.</description>
    </extension>
    <extension alias="os-user-data" updated="2012-08-07T00:00:00+00:00"
      namespace="http://docs.openstack.org/compute/ext/userdata/api/v1.1"
      name="UserData">
      <description>Add user_data to the Create Server v1.1
```

```
    API.</description>
</extension>
<extension alias="os-user-quotas"
  updated="2013-07-18T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/user_quotas/api/v1.1"
  name="UserQuotas">
  <description>Project user quota support.</description>
</extension>
<extension alias="os-virtual-interfaces"
  updated="2011-08-17T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/virtual_interfaces/api/
v1.1"
  name="VirtualInterfaces">
  <description>Virtual interface support.</description>
</extension>
<extension alias="os-volume-attachment-update"
  updated="2013-06-20T00:00:00-00:00"
  namespace="http://docs.openstack.org/compute/ext/os-volume-attachment-
update/api/v2"
  name="VolumeAttachmentUpdate">
  <description>Support for updating a volume
    attachment.</description>
</extension>
<extension alias="os-volumes" updated="2011-03-25T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/volumes/api/v1.1"
  name="Volumes">
  <description>Volumes support.</description>
</extension>
<extension alias="os-extended-services-delete"
  updated="2013-12-10T00:00:00"
  namespace="http://docs.openstack.org/compute/ext/extended_services_delete/
api/v2"
  name="ExtendedServicesDelete">
  <description>Services deletion support.</description>
</extension>
<extension alias="os-console-auth-tokens"
  updated="2013-08-13T00:00:00+00:00"
  namespace="http://docs.openstack.org/compute/ext/consoles-auth-tokens/api/
v2"
  name="ConsoleAuthTokens">
  <description>Console token authentication support.</description>
</extension>
</extensions>
```

### 3.3.2. Get extension

Method	URI	Description
GET	/v2/{tenant_id}/extensions/{alias}	Gets details about the specified extension.

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

**Normal response codes:** 200, 203

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

#### 3.3.2.1. Request

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

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

#### 3.3.2.2. Response

##### Example 3.9. 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 3.10. Get extension: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extension xmlns:atom="http://www.w3.org/2005/Atom"
           xmlns="http://docs.openstack.org/common/api/v1.0" alias="OS-DCF"
           updated="2011-09-27T00:00:00+00:00"
           namespace="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
           name="DiskConfig">
  <description>Disk Management Extension.</description>
</extension>
```

## 3.4. Servers

List, create, get details for, update, and delete servers.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/servers{?changes-since,image,flavor,name,marker,limit,status,host}	Lists IDs, names, and links for all servers.
<b>POST</b>	/v2/{tenant_id}/servers	Creates a server.
<b>GET</b>	/v2/{tenant_id}/servers/detail{?changes-since,image,flavor,name,marker,limit,status,host}	Lists details for all servers.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}	Gets details for a specified server.
<b>PUT</b>	/v2/{tenant_id}/servers/{server_id}	Updates the editable attributes of the specified server.
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}	Deletes a specified server.

### 3.4.1. List servers

Method	URI	Description
GET	/v2/{tenant_id}/servers{?changes-since,image,flavor,name,marker,limit,status,host}	Lists IDs, names, and links for all servers.

**Normal response codes:** 200, 203

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

#### 3.4.1.1. 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	DateTime <i>(Optional)</i>	A time/date stamp for when the server last changed status.
image	AnyURI <i>(Optional)</i>	Name of the image in URL format.
flavor	AnyURI <i>(Optional)</i>	Name of the flavor in URL format.
name	String <i>(Optional)</i>	Name of the server as a string.
marker	UUID <i>(Optional)</i>	UUID of the server at which you want to set a marker.
limit	Int <i>(Optional)</i>	Integer value for the limit of values to return.
status	ServerStatus <i>(Optional)</i>	Value of the status of the server so that you can filter on "ACTIVE" for example.
host	String <i>(Optional)</i>	Name of the host as a string.

This operation does not accept a request body.

#### 3.4.1.2. Response

##### Example 3.11. List servers: JSON response

```
{
```

```

"servers": [
    {
        "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/
servers/616fb98f-46ca-475e-917e-2563e5a8cd19",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
616fb98f-46ca-475e-917e-2563e5a8cd19",
                "rel": "bookmark"
            }
        ],
        "name": "new-server-test"
    }
]
}

```

This table shows the body parameters for the list servers response:

Name	Type	Description
servers	Servers WithOnly IDsNames Links  <i>(Required)</i>	List of servers.
next	AnyURI  <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI  <i>(Optional)</i>	Moves to the previous metadata item.

### Example 3.12. List servers: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
    <server name="new-server-test" id="b626796d-d585-4874-b178-78c65289bba4">
        <atom:link href="http://openstack.example.com/v2/openstack/servers/
b626796d-d585-4874-b178-78c65289bba4" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/servers/b626796d-
d585-4874-b178-78c65289bba4" rel="bookmark"/>
    </server>
</servers>

```

## 3.4.2. Create server

Method	URI	Description
POST	/v2/{tenant_id}/servers	Creates a server.

**Normal response codes:** 202

**Error response codes:** computeFault (400, 500, ...), UnprocessableEntity (422), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), serverCapacityUnavailable (503)

### 3.4.2.1. 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.

#### Example 3.13. Create server: JSON request

```
{
  "server": {
    "name": "server-test-1",
    "imageRef": "b5660a6e-4b46-4be3-9707-6b47221b454f",
    "flavorRef": "2",
    "max_count": 1,
    "min_count": 1,
    "networks": [
      {
        "uuid": "d32019d3-bc6e-4319-9c1d-6722fc136a22"
      }
    ],
    "security_groups": [
      {
        "name": "default"
      },
      {
        "name": "another-secgroup-name"
      }
    ]
  }
}
```

#### Example 3.14. Create server: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  imageRef="b5660a6e-4b46-4be3-9707-6b47221b454f" flavorRef="2"
  name="server-test-2">
  <metadata>
    <meta key="My Server Name">API Test Server XML</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
```

```

ICAgICAgDQoIQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCbtb3Z1cyBpbIBqdXN0IHN1Y2ggYSBkaXJ1Y3Rpb24gYW5k
IGF0IHN1Y2ggYSBzcGV1ZC4uLk10IGZlZWxzIGFuIGltcHVs
c21vbi4uLnRoaXMgaXMgdGh1IHBsYWN1IHRvIGdvIG5vdy4g
QnV0IHRoZSBza3kg a25vd3MgdGh1IHZjYXNvb nMgYW5kIHRo
ZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3Vkc ywgYW5kIHZj
dSB3aWxsIGtub3csIHRvbywd2h1biB5b3UgbG1mdCB5b3VY
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
b25zLiINCg0KLVJpY2hhcmQgQmFjaA==</file>
</personality>
<networks>
<network uuid="0ef47ac7-6797-4e01-8a47-ed26ec3aaa56" />
</networks>
<security_groups>
<security_group name="default" />
<security_group name="another-secgroup-name" />
</security_groups>
</server>

```

### 3.4.2.2. Response

#### Example 3.15. Create server: JSON response

```
{
  "server": {
    "security_groups": [
      {
        "name": "default"
      }
    ],
    "OS-DCF:diskConfig": "MANUAL",
    "id": "c6d04159-9bfcc-4ab8-823d-0d5ca2abe152",
    "links": [
      {
        "href": "http://166.78.46.130:8774/v2/4fd44f30292945e481c7b8a0c8908869/servers/c6d04159-9bfcc-4ab8-823d-0d5ca2abe152",
        "rel": "self"
      },
      {
        "href": "http://166.78.46.130:8774/4fd44f30292945e481c7b8a0c8908869/servers/c6d04159-9bfcc-4ab8-823d-0d5ca2abe152",
        "rel": "bookmark"
      }
    ],
    "adminPass": "aabbccddeeff"
  }
}
```

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

Name	Type	Description
security_groups	String <i>(Optional)</i>	A security_groups object.
security_group	String <i>(Optional)</i>	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.

Name	Type	Description
user_data	String <i>(Optional)</i>	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String <i>(Optional)</i>	The availability zone in which to launch the server.
server	ServerFor Create <i>(Required)</i>	server.
imageRef	String <i>(Required)</i>	The image reference for the desired image for your server instance. Specify as an ID or full URL.
flavorRef	String <i>(Required)</i>	The flavor reference for the desired flavor for your server instance. Specify as an ID or full URL.
networks	String <i>(Optional)</i>	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant.  Optionally, you can create one or more NICs on the server.  To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a networks object.  To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a networks object.  You can specify multiple NICs on the server.
uuid	String <i>(Optional)</i>	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a networks object. Required if you omit the <code>port</code> attribute.
port	String <i>(Optional)</i>	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a networks object. Required if you omit the <code>uuid</code> attribute.
fixed_ip	String <i>(Optional)</i>	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
name	String <i>(Required)</i>	The server name.
metadata	String <i>(Optional)</i>	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
personality	String <i>(Optional)</i>	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
config_drive	String <i>(Optional)</i>	Enables metadata injection in a server through a configuration drive. To enable a configuration drive, specify <code>true</code> . Otherwise, specify <code>false</code> .

### Example 3.16. Create server: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server
  xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  id="9720d338-07fb-4317-9eda-a0a833226afa" adminPass="aabbccddeeff"
  OS-DCF:diskConfig="MANUAL">
```

```
<metadata/>
<atom:link
  href="http://166.78.46.130:8774/v2/4fd44f30292945e481c7b8a0c8908869/
servers/9720d338-07fb-4317-9eda-a0a833226afa"
  rel="self"/>
<atom:link
  href="http://166.78.46.130:8774/4fd44f30292945e481c7b8a0c8908869/servers/
9720d338-07fb-4317-9eda-a0a833226afa"
  rel="bookmark"/>
<security_groups>
  <security_group name="default" />
</security_groups>
</server>
```

### 3.4.3. List details for servers

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

**Normal response codes:** 200, 203

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

#### 3.4.3.1. 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 <i>(Optional)</i>	A time/date stamp for when the server last changed status.
image	AnyURI <i>(Optional)</i>	Name of the image in URL format.
flavor	AnyURI <i>(Optional)</i>	Name of the flavor in URL format.
name	String <i>(Optional)</i>	Name of the server as a string.
marker	UUID <i>(Optional)</i>	UUID of the server at which you want to set a marker.
limit	Int <i>(Optional)</i>	Integer value for the limit of values to return.
status	ServerStatus <i>(Optional)</i>	Value of the status of the server so that you can filter on "ACTIVE" for example.
host	String <i>(Optional)</i>	Name of the host as a string.

This operation does not accept a request body.

#### 3.4.3.2. Response

This table shows the body parameters for the list details for servers response:

Name	Type	Description
next <i>(Optional)</i>	AnyURI	Moves to the next metadata item.
previous <i>(Optional)</i>	AnyURI	Moves to the previous metadata item.

### Example 3.17. List details for servers: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2012-09-07T17:11:46Z" hostId="1b3afbf40723a9649091142a647b83eb6e5b49973239cdeb3d1973c" name="new-server-test" created="2012-09-07T17:11:45Z" userId="fake" tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id="7c3c0f79-343e-4b99-93bc-2ade47641e64">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/openstack/flavors/1" rel="bookmark"/>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" addr="192.168.0.3"/>
            </network>
        </addresses>
        <atom:link href="http://openstack.example.com/v2/openstack/servers/7c3c0f79-343e-4b99-93bc-2ade47641e64" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/servers/7c3c0f79-343e-4b99-93bc-2ade47641e64" rel="bookmark"/>
    </server>
</servers>
```

### Example 3.18. List details for servers: JSON response

```
{
    "servers": [
        {
            "accessIPv4": "",
            "accessIPv6": "",
            "addresses": {
                "private": [
                    {
                        "addr": "192.168.0.3",
                        "version": 4
                    }
                ]
            },
            "created": "2012-09-07T16:56:37Z",
            "flavor": {
                "id": "1",
                "links": [

```

```

        {
            "href": "http://openstack.example.com/openstack/
flavors/1",
            "rel": "bookmark"
        }
    ],
},
"hostId":
"16d193736a5cfdb60c697ca27ad071d6126fa13baeb670fc9d10645e",
"id": "05184ba3-00ba-4fbc-b7a2-03b62b884931",
"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/05184ba3-00ba-4fbc-b7a2-03b62b884931",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/servers/
05184ba3-00ba-4fbc-b7a2-03b62b884931",
            "rel": "bookmark"
        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2012-09-07T16:56:37Z",
    "user_id": "fake"
}
]
}

```

### Example 3.19. List details for servers: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2012-09-07T17:11:46Z" hostId=
"1b3afbfff40723a9649091142a647b83eb6e5b49973239cdeb3d1973c" name=
"new-server-test" created="2012-09-07T17:11:45Z" userId="fake"
    tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id=
"7c3c0f79-343e-4b99-93bc-2ade47641e64">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">

```

```
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel="bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" addr="192.168.0.3"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/7c3c0f79-343e-4b99-93bc-2ade47641e64" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/7c3c0f79-343e-4b99-93bc-2ade47641e64" rel="bookmark"/>
    </server>
</servers>
```

This operation does not return a response body.

## 3.4.4. Get server details

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}	Gets details for a specified server.

**Normal response codes:** 200, 203

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

### 3.4.4.1. 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.

### 3.4.4.2. Response

#### Example 3.20. Get server details: JSON response

```
{
  "server": {
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": [
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    ],
    "created": "2012-08-20T21:11:09Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "65201c14a29663e06d0748e561207d998b343e1d164bfa0aafa9c45d",
    "id": "893c7791-f1df-4c3d-8383-3caae9656c62",
    "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/
893c7791-f1df-4c3d-8383-3caae9656c62",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/servers/
893c7791-f1df-4c3d-8383-3caae9656c62",
            "rel": "bookmark"
        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2012-08-20T21:11:09Z",
    "user_id": "fake"
}
}

```

### Example 3.21. Get server details: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1" status="ACTIVE" updated="2012-08-20T21:11:10Z" hostId=
"1746536de20daadad89a6fab8d6968b1214b0ba9fb37b29e7098e0b9" name="new-server-
test" created="2012-08-20T21:11:10Z" userId="fake" tenantId="openstack"
accessIPv4="" accessIPv6="" progress="0" id="3f9f7d18-aaf3-4703-b368-
ea9b4d609c95">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" addr="192.168.0.3"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/3f9f7d18-
aa9b4d609c95" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/3f9f7d18-
aa9b4d609c95" rel="bookmark"/>

```

```
</server>
```

## 3.4.5. Update server

Method	URI	Description
PUT	/v2/{tenant_id}/servers/{server_id}	Updates the editable attributes of the specified server.

**Normal response codes:** 200

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

### 3.4.5.1. 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 3.22. Update Server Name Request: JSON

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

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

Name	Type	Description
name	ServerFor Update <i>(Optional)</i>	The name of the server. If you edit the server name, the server host name does not change. Also, server names are not guaranteed to be unique.
accessIPv4	ServerFor Update <i>(Optional)</i>	The IP version 4 address.
accessIPv6	ServerFor Update <i>(Optional)</i>	The IP version 6 address.

#### Example 3.23. Update Server Name Request: XML

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

#### Example 3.24. Update Server IP Addresses Request: JSON

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

```
}
```

### Example 3.25. Update Server IP Addresses Request: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<server
    xmlns="http://docs.openstack.org/compute/api/v1.1"
    accessIPv4="67.23.10.132"
    accessIPv6="::babe:67.23.10.132"
/>
```

This operation does not accept a request body.

### 3.4.5.2. Response

#### Example 3.26. 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",
        "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 3.27. Update Server Name Response: XML

```

<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
         xmlns:atom="http://www.w3.org/2005/Atom"
         id="52415800-8b69-11e0-9b19-734f565bc83b"
         tenant_id="1234" user_id="5678"
         name="new-server-test"
         hostId="e4d909c290d0fb1ca068ffaddf22cbd0" progress="0"
         status="ACTIVE"
         created="2010-11-11T12:00:00Z"
         updated="2010-11-12T12:44:44Z"
         accessIPv4="67.23.10.138"
         accessIPv6="::babe:67.23.10.138">
    <image id="52415800-8b69-11e0-9b19-734f6f006e54"
           name="CentOS 5.2">
        <atom:link
            rel="self"

```

```

        href="http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"/>
    <atom:link
        rel="bookmark"
        href="http://servers.api.openstack.org/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"/>
    </image>
    <flavor id="52415800-8b69-11e0-9b19-734f1195ff37"
            name="256 MB Server">
        <atom:link
            rel="self"
            href="http://servers.api.openstack.org/v2/1234/flavors/
52415800-8b69-11e0-9b19-734f1195ff37"/>
        <atom:link
            rel="bookmark"
            href="http://servers.api.openstack.org/1234/flavors/
52415800-8b69-11e0-9b19-734f1195ff37"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="public">
            <ip version="4" addr="67.23.10.138"/>
            <ip version="6" addr="::babe:67.23.10.138"/>
        </network>
        <network id="private">
            <ip version="4" addr="10.176.42.19"/>
            <ip version="6" addr="::babe:10.176.42.19"/>
        </network>
    </addresses>
    <atom:link
        rel="self"
        href="http://servers.api.openstack.org/v2/1234/servers/
52415800-8b69-11e0-9b19-734fcece0043"/>
    <atom:link
        rel="bookmark"
        href="http://servers.api.openstack.org/1234/servers/
52415800-8b69-11e0-9b19-734fcece0043"/>
</server>
```

### Example 3.28. 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"
    }
]
}
}
```

### Example 3.29. Update Server IP Addresses Response: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
         xmlns:atom="http://www.w3.org/2005/Atom"
         id="52415800-8b69-11e0-9b19-734f565bc83b"
         tenant_id="1234" user_id="5678"
         name="new-server-test"
         hostId="e4d909c290d0fb1ca068ffaddf22cbd0" progress="0"
         status="ACTIVE"
         created="2010-11-11T12:00:00Z"
         updated="2010-11-12T12:55:55Z"
         accessIPv4="67.23.10.132"
         accessIPv6="::babe:67.23.10.132">
<image id="52415800-8b69-11e0-9b19-734f6f006e54"
       name="CentOS 5.2">
    <atom:link
        rel="self"
        href="http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"/>
    <atom:link
        rel="bookmark"
        href="http://servers.api.openstack.org/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"/>
</image>
<flavor id="52415800-8b69-11e0-9b19-734f1195ff37"
       name="256 MB Server">
    <atom:link
        rel="self"
        href="http://servers.api.openstack.org/v2/1234/flavors/
52415800-8b69-11e0-9b19-734f1195ff37"/>
    <atom:link
        rel="bookmark"
        href="http://servers.api.openstack.org/1234/flavors/
52415800-8b69-11e0-9b19-734f1195ff37"/>
</flavor>
<metadata>
    <meta key="My Server Name">Apache1</meta>
</metadata>
<addresses>
    <network id="public">
        <ip version="4" addr="67.23.10.138"/>
        <ip version="6" addr="::babe:67.23.10.138"/>
    </network>
    <network id="private">
        <ip version="4" addr="10.176.42.19"/>
        <ip version="6" addr="::babe:10.176.42.19"/>
    </network>
</addresses>
<atom:link
    rel="self"
```

```
    href="http://servers.api.openstack.org/v2/1234/servers/
52415800-8b69-11e0-9b19-734fcece0043"/>
<atom:link
  rel="bookmark"
  href="http://servers.api.openstack.org/1234/servers/
52415800-8b69-11e0-9b19-734fcece0043"/>
</server>
```

This operation does not return a response body.

## 3.4.6. Delete server

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}	Deletes a specified server.

**Normal response codes:** 204

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

### 3.4.6.1. 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.

## 3.5. Server metadata

Show details for, set, update, and delete server metadata or metadata items.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/metadata	Shows metadata for a specified server.
<b>PUT</b>	/v2/{tenant_id}/servers/{server_id}/metadata	Creates or replaces metadata for a specified server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/metadata	Updates metadata items by key for a specified server.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/metadata/{key}	Shows details for a metadata item by key for a specified server.
<b>PUT</b>	/v2/{tenant_id}/servers/{server_id}/metadata/{key}	Sets a metadata item by key for a specified server.
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}/metadata/{key}	Deletes a metadata item by key for a specified server.

## 3.5.1. Show server metadata

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/metadata	Shows metadata for a specified server.

**Normal response codes:** 200, 203

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

### 3.5.1.1. Request

This table shows the URI parameters for the show server 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.

### 3.5.1.2. Response

#### Example 3.30. Show server metadata: JSON response

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

This table shows the body parameters for the show server metadata response:

Name	Type	Description
next	AnyURI <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI <i>(Optional)</i>	Moves to the previous metadata item.

#### Example 3.31. Show server metadata: XML response

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

### 3.5.2. Create or replace server metadata

Method	URI	Description
PUT	/v2/{tenant_id}/servers/{server_id}/metadata	Creates or replaces metadata for a specified server.

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

If the number of metadata items exceeds the quota for metadata items, an overLimit (413) fault might be thrown.

**Normal response codes:** 200

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

#### 3.5.2.1. Request

This table shows the URI parameters for the create or replace server metadata 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. Create or replace server metadata: JSON request

```
{
  "metadata": {
    "name": "test_server"
  }
}
```

#### Example 3.33. Create or replace server metadata: XML request

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

#### 3.5.2.2. Response

#### Example 3.34. Create or replace server metadata: JSON response

```
{
  "metadata": {
    "name": "test_server",
    "server_type": "test"
  }
}
```

This table shows the body parameters for the create or replace server metadata response:

Name	Type	Description
next	AnyURI <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI <i>(Optional)</i>	Moves to the previous metadata item.

**Example 3.35. Create or replace server metadata: XML response**

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

### 3.5.3. Update server metadata items

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/metadata	Updates metadata items by key for a specified server.

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

An overLimit (413) fault might be thrown if the operation causes the quota for metadata items to be exceeded.

**Normal response codes:** 200

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

#### 3.5.3.1. Request

This table shows the URI parameters for the update server 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 3.36. Update server metadata items: JSON request

```
{
  "metadata": {
    "name": "test_server"
  }
}
```

#### Example 3.37. Update server metadata items: XML request

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

#### 3.5.3.2. Response

#### Example 3.38. Update server metadata items: JSON response

```
{
  "metadata": {
    "name": "test_server",
    "server_type": "test"
  }
}
```

This table shows the body parameters for the update server metadata items response:

Name	Type	Description
next	AnyURI <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI <i>(Optional)</i>	Moves to the previous metadata item.

**Example 3.39. Update server metadata items: XML response**

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

## 3.5.4. Show server metadata item details

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/metadata/{key}	Shows details for a metadata item by key for a specified server.

**Normal response codes:** 200, 203

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

### 3.5.4.1. Request

This table shows the URI parameters for the show server 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}	Metadata Key	A string. Maximum length is 255 characters.

This operation does not accept a request body.

### 3.5.4.2. Response

#### Example 3.40. Show server metadata item details: JSON response

```
{
    "metadata": {
        "name": "test_server",
        "server_type": "test"
    }
}
```

#### Example 3.41. Show server metadata item details: XML response

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

### 3.5.5. Create or update server metadata item

Method	URI	Description
PUT	/v2/{tenant_id}/servers/{server_id}/metadata/{key}	Sets a metadata item by key for a specified server.

An overLimit (413) fault might be thrown if the operation causes the quota for metadata items to be exceeded.

**Normal response codes:** 200

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

#### 3.5.5.1. Request

This table shows the URI parameters for the create or update server 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}	Metadata Key	A string. Maximum length is 255 characters.

#### Example 3.42. Create or update server metadata item: JSON request

```
{
    "metadata": {
        "name": "test_server"
    }
}
```

#### Example 3.43. Create or update server metadata item: XML request

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

#### 3.5.5.2. Response

#### Example 3.44. Create or update server metadata item: JSON response

```
{
    "metadata": {
        "name": "test_server",
        "server_type": "test"
    }
}
```

#### Example 3.45. Create or update server metadata item: XML response

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

```
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
    <meta key="type">staging_server</meta>
</metadata>
```

## 3.5.6. Delete server metadata item

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}/metadata/{key}	Deletes a metadata item by key for a specified server.

**Normal response codes:** 204

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), buildInProgress (409)

### 3.5.6.1. Request

This table shows the URI parameters for the delete server 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}	Metadata Key	A string. Maximum length is 255 characters.

This operation does not accept a request body.

## 3.6. Server addresses

List addresses for a specified server or a specified server and network.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/ips	Lists networks and addresses for a specified tenant and server.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/ips/{network_label}	Lists addresses for a specified tenant, server, and network.

## 3.6.1. List addresses

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/ips	Lists networks and addresses for a specified tenant and server.

Specify the tenant ID and server ID in the URI.

**Normal response codes:** 200, 203

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), buildInProgress (409)

### 3.6.1.1. Request

This table shows the URI parameters for the list addresses 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 3.46. List addresses: JSON response

```
{
  "addresses": {
    "public": [
      {
        "version": 4,
        "addr": "67.23.10.132"
      },
      {
        "version": 6,
        "addr": "::babe:67.23.10.132"
      },
      {
        "version": 4,
        "addr": "67.23.10.131"
      },
      {
        "version": 6,
        "addr": "::babe:4317:0A83"
      }
    ],
    "private": [
      {
        "version": 4,
        "addr": "10.176.42.16"
      },
      {
        "version": 6,
        "addr": "::babe:4317:0A83"
      }
    ]
  }
}
```

```
        "version": 6,
        "addr": "::babe:10.176.42.16"
    }
]
}
```

### Example 3.47. List addresses: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<addresses xmlns="http://docs.openstack.org/compute/api/v1.1">
    <network id="public">
        <ip version="4" addr="67.23.10.132"/>
        <ip version="6" addr="::babe:67.23.10.132"/>
        <ip version="4" addr="67.23.10.131"/>
        <ip version="6" addr="::babe:4317:0A83"/>
    </network>
    <network id="private">
        <ip version="4" addr="10.176.42.16"/>
        <ip version="6" addr="::babe:10.176.42.16"/>
    </network>
</addresses>
```

## 3.6.2. List addresses by network

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/ips/{network_label}	Lists addresses for a specified tenant, server, and network.

Specify the tenant ID, server ID, and network label in the request URI.

**Normal response codes:** 200, 203

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), buildInProgress (409)

### 3.6.2.1. Request

This table shows the URI parameters for the list addresses by network 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 public or private.

This operation does not accept a request body.

### 3.6.2.2. Response

#### Example 3.48. List addresses by network: JSON response

```
{
  "network": {
    "id": "public",
    "ip": [
      {
        "version": 4,
        "addr": "67.23.10.132"
      },
      {
        "version": 6,
        "addr": "::babe:67.23.10.132"
      },
      {
        "version": 4,
        "addr": "67.23.10.131"
      },
      {
        "version": 6,
        "addr": "::babe:4317:0A83"
      }
    ]
  }
}
```

### Example 3.49. List addresses by network: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<network xmlns="http://docs.openstack.org/compute/api/v1.1"
          id="public">
    <ip version="4" addr="67.23.10.132"/>
    <ip version="6" addr="::babe:67.23.10.132"/>
    <ip version="4" addr="67.23.10.131"/>
    <ip version="6" addr="::babe:4317:0A83"/>
</network>
```

## 3.7. Server actions

Perform actions for a specified server, including change administrator password, reboot, rebuild, resize, and create image from server.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Changes the password for a server. Specify the <code>changePassword</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Reboots the specified server. Specify the <code>reboot</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Rebuilds the specified server. Specify the <code>rebuild</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Resizes the specified server. Specify the <code>resize</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Confirms a pending resize action. Specify the <code>confirmResize</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Cancels and reverts a pending resize action. Specify the <code>revertResize</code> action in the request body.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Creates a new image. Specify the <code>createImage</code> action in the request body.

## 3.7.1. Change password

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Changes the password for a server. Specify the changePassword 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), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

### 3.7.1.1. Request

This table shows the URI parameters for the change password 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. Change password: JSON request

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

#### Example 3.51. Change password: XML request

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

## 3.7.2. Reboot server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Reboots the specified 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), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), HTTPUnprocessableEntity (422), buildInProgress (409)

### 3.7.2.1. Request

This table shows the URI parameters for the reboot 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 3.52. Reboot server: JSON request

```
{
    "reboot": {
        "type": "SOFT"
    }
}
```

#### Example 3.53. Reboot server: XML request

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

### 3.7.3. Rebuild server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Rebuilds the specified server. Specify the rebuild 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), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), serverCapacityUnavailable (503), buildInProgress (409)

#### 3.7.3.1. Request

This table shows the URI parameters for the rebuild 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 3.54. Rebuild server: JSON request

```
{
    "rebuild": {
        "imageRef": "http://openstack.example.com/v2/32278/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "name": "foobar",
        "adminPass": "seekr3t",
        "accessIPv4": "1.2.3.4",
        "accessIPv6": "fe80::100",
        "metadata": {
            "meta var": "meta val"
        },
        "personality": [
            {
                "path": "/etc/banner.txt",
                "contents": "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCBtb3Z1cyBpbIBqdXN0IHN1Y2ggYSBkaXJ1Y3Rp24gYW5k
IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZ1ZWxzIGFuIGltcHVs
c21vb14uLnRoaXMgaXMgdGh1IHBsYWN1IHRvIGdvIG5vdy4g
QnV0IHRoZSBza3kg25vd3MgdGh1IHZ1YXNvbnMgYW5kIHRo
ZSBwYXR0ZXJuicyBiZWhpbmQgYWxsIGNsb3VkcwygYW5kIHZ1
dSB3aWxsIGtub3csIHRvbywg2h1biB5b3UgbG1mdCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
b25zLiINCg0KLVJpY2hhcmQgQmFjaA=="
            }
        ]
    }
}
```

#### Example 3.55. Rebuild server: XML request

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

```

        name="foobar"
        imageRef="http://openstack.example.com/v1.1/32278/images/
70a599e0-31e7-49b7-b260-868f441e862b"
        accessIPv4="1.2.3.4"
        accessIPv6="fe80::100"
        adminPass="seekr3t">
<metadata>
    <meta key="My Server Name">Apache1</meta>
</metadata>
<personality>
    <file path="/etc/banner.txt">
        ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
        dCBtb3ZlcYBpbBqdxN0IHN1Y2ggYSBkaXJ1Y3RpB24gYW5k
        IGF0IHN1Y2ggYSBzCGV1ZC4uLk10IGZ1ZWxzIGFuIGltcHVs
        c2lvbi4uLnRoaXMgaXMgdGh1IHByWN1IHRvIGdvIG5vdy4g
        QnV0IHRoZSBza3kga25vd3MgdGh1IHJ1YXNvbMgYW5kiHRo
        ZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3VkcwYgYW5kiHl
        dSB3aWxsIGtub3csIHRvbywgd2h1biB5b3UgbG1mdCB5b3V
        c2VsZiBoawdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
        b25zLiINCg0KLVJpY2hhcmQgQmFjaA==
    </file>
</personality>
</rebuild>

```

### 3.7.3.2. Response

#### Example 3.56. Rebuild server: JSON response

```
{
    "server": {
        "accessIPv4": "1.2.3.4",
        "accessIPv6": "fe80::100",
        "addresses": {
            "private": [
                {
                    "addr": "192.168.0.3",
                    "version": 4
                }
            ]
        },
        "adminPass": "seekr3t",
        "created": "2012-09-12T17:20:36Z",
        "flavor": {
            "id": "1",
            "links": [
                {
                    "href": "http://openstack.example.com/openstack/flavors/
1",
                    "rel": "bookmark"
                }
            ]
        },
        "hostId": "1e3da81662354c25560b7e5ea6d8123031f67168b6992f20bb84df69",
        "id": "075e40fe-9f03-4652-ba8e-5f8e2547899a",
        "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/
075e40fe-9f03-4652-ba8e-5f8e2547899a",
    "rel": "self"
},
{
    "href": "http://openstack.example.com/openstack/servers/
075e40fe-9f03-4652-ba8e-5f8e2547899a",
    "rel": "bookmark"
}
],
"metadata": {
    "meta var": "meta val"
},
"name": "foobar",
"progress": 0,
"status": "ACTIVE",
"tenant_id": "openstack",
"updated": "2012-09-12T17:20:37Z",
"user_id": "fake"
}
}
}

```

### Example 3.57. Rebuild server: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1" status="ACTIVE" updated="2012-09-14T16:41:46Z" hostId=
"a0e37e3bd9f674600aabecbc123d80ae2717ace90893d79cd4abc46" name="foobar"
created="2012-09-14T16:41:45Z" userId="fake" tenantId="openstack" accessIPv4=
"1.2.3.4" accessIPv6="fe80::100" progress="0" id="943acea5-2fc8-4f31-
bab6-8f7b9ac923ca" adminPass="seekr3t">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" addr="192.168.0.3"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/
943acea5-2fc8-4f31-bab6-8f7b9ac923ca" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/
943acea5-2fc8-4f31-bab6-8f7b9ac923ca" rel="bookmark"/>
</server>

```

## 3.7.4. Resize server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Resizes the specified server. Specify the <code>resize</code> 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), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), serverCapacityUnavailable (503), buildInProgress (409)

### 3.7.4.1. Request

This table shows the URI parameters for the resize 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 3.58. Resize server: JSON request

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

#### Example 3.59. Resize server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<resize
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  flavorRef="2"/>
```

## 3.7.5. Confirm resized server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Confirms a pending resize action. Specify the confirm-Resize action in the request body.

**Normal response codes:** 204

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), serverCapacityUnavailable (503), buildInProgress (409)

### 3.7.5.1. Request

This table shows the URI parameters for the confirm resized 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 3.60. Confirm resized server: JSON request

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

#### Example 3.61. Confirm resized server: XML request

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

## 3.7.6. Revert resized server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Cancels and reverts a pending resize action. Specify the revertResize 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), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), serverCapacityUnavailable (503), buildInProgress (409)

### 3.7.6.1. Request

This table shows the URI parameters for the revert resized 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 3.62. Revert resized server: JSON request

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

#### Example 3.63. Revert resized server: XML request

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

## 3.7.7. Create image

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Creates a new image. Specify the <code>createImage</code> action in the request body.

**Normal response codes:** 202

**Error response codes:** computeFault (400, 500, ...), backupOrResizelnProgress (409), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), serverCapacityUnavailable (503), buildInProgress (409)

### 3.7.7.1. Request

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

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

#### Example 3.64. Create image: JSON request

```
{
  "createImage": {
    "name": "foo-image",
    "metadata": {
      "myvar": "foobar"
    }
  }
}
```

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

Name	Type	Description
name	ServerFor Create <i>(Required)</i>	Name of the snapshot.
createImage	ServerFor Create <i>(Required)</i>	A dictionary representation of a creation for image.
metadata	ServerFor Create <i>(Optional)</i>	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.

#### Example 3.65. Create image: XML request

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

```
<meta key="myvar">foobar</meta>
</metadata>
</createImage>
```

### 3.7.7.2. Response

This operation does not return a response body.

## 3.8. Flavors

List available flavors and get details for a specified flavor. A flavor is a hardware configuration for a server. Each flavor is a unique combination of disk space and memory capacity.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/flavors{?minDisk,minRam,marker,limit}	Lists IDs, names, and links for available flavors.
<b>GET</b>	/v2/{tenant_id}/flavors/detail{?minDisk,minRam,marker,limit}	Lists all details for available flavors.
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor.

## 3.8.1. List flavors

Method	URI	Description
GET	/v2/{tenant_id}/flavors{?minDisk,minRam,marker,limit}	Lists IDs, names, and links for available flavors.

**Normal response codes:** 200, 203

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

### 3.8.1.1. 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.

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

Name	Type	Description
minDisk	Int <i>(Optional)</i>	Integer value for the minimum disk space in GB so you can filter results.
minRam	Int <i>(Optional)</i>	Integer value for the minimum RAM so you can filter results.
marker	UUID <i>(Optional)</i>	UUID of the flavor at which you want to set a marker.
limit	Int <i>(Optional)</i>	Integer value for the limit of values to return.

This operation does not accept a request body.

### 3.8.1.2. Response

#### Example 3.66. 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"
        }
    ]
}
```

```

        "rel": "bookmark"
    }
],
"name": "m1.xlarge"
}
]
}

```

This table shows the body parameters for the list flavors response:

Name	Type	Description
flavors	Flavors WithOnly IDsNames Links  <i>(Required)</i>	Flavors are known combinations of memory, disk space, and number of CPUs.
next	AnyURI  <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI  <i>(Optional)</i>	Moves to the previous metadata item.

### Example 3.67. List flavors: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
    <flavor name="m1.tiny" id="1">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <flavor name="m1.small" id="2">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/2" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/2" rel=
"bookmark"/>
    </flavor>
    <flavor name="m1.medium" id="3">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/3" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/3" rel=
"bookmark"/>
    </flavor>
    <flavor name="m1.large" id="4">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/4" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/4" rel=
"bookmark"/>
    </flavor>
    <flavor name="m1.xlarge" id="5">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/5" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/5" rel=
"bookmark"/>
    </flavor>
</flavors>

```



## 3.8.2. List details for flavors

Method	URI	Description
GET	/v2/{tenant_id}/flavors/detail{?minDisk,minRam,marker,limit}	Lists all details for available flavors.

**Normal response codes:** 200, 203

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

### 3.8.2.1. Request

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

Name	Type	Description
minDisk	Int <i>(Optional)</i>	Integer value for the minimum disk space in GB so you can filter results.
minRam	Int <i>(Optional)</i>	Integer value for the minimum RAM so you can filter results.
marker	UUID <i>(Optional)</i>	UUID of the flavor at which you want to set a marker.
limit	Int <i>(Optional)</i>	Integer value for the limit of values to return.

This operation does not accept a request body.

### 3.8.2.2. Response

#### Example 3.68. List details for 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"
        }
    ]
}
```

```

        "rel": "bookmark"
    }
],
"name": "m1.xlarge"
}
]
}

```

This table shows the body parameters for the list details for flavors response:

Name	Type	Description
next	AnyURI <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI <i>(Optional)</i>	Moves to the previous metadata item.

### Example 3.69. List details for flavors: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
    <flavor name="m1.tiny" id="1">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <flavor name="m1.small" id="2">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/2" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/2" rel=
"bookmark"/>
    </flavor>
    <flavor name="m1.medium" id="3">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/3" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/3" rel=
"bookmark"/>
    </flavor>
    <flavor name="m1.large" id="4">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/4" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/4" rel=
"bookmark"/>
    </flavor>
    <flavor name="m1.xlarge" id="5">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/5" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/5" rel=
"bookmark"/>
    </flavor>
</flavors>

```

### 3.8.3. Get flavor details

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

**Normal response codes:** 200, 203

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

#### 3.8.3.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{flavor_id}	UUID	UUID for the specific flavor (combination of memory, disk size, and CPUs).

This operation does not accept a request body.

#### 3.8.3.2. Response

##### Example 3.70. Get flavor details: JSON response

```
{
  "flavor": {
    "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",
    "ram": 512,
    "vcpus": 1
  }
}
```

##### Example 3.71. Get flavor details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" disk="1" vcpus="1" ram="512" name="m1.tiny" id="1">
  <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel="self"/>
```

```
<atom:link href="http://openstack.example.com/openstack/flavors/1" rel="bookmark"/>
</flavor>
```

## 3.9. Images

List available images, get details for a specified image, and delete an image.

Also, set, list, get details for, and delete image metadata.

An image is a collection of files that you use to create or rebuild a server. By default, operators provide pre-built operating system images. You can also create custom images: See [Section 3.7, "Server actions" \[172\]](#).

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

## 3.9.1. List images

Method	URI	Description
GET	/v2/{tenant_id}/images{?changes-since,server,name,status,marker,limit,type}	Lists IDs, names, and links for available images.

**Normal response codes:** 200, 203

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

### 3.9.1.1. Request

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

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

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

Name	Type	Description
changes-since	DateTime (Optional)	A time/date stamp for when the image last changed status.
server	AnyURI (Optional)	Name of the server in URL format.
name	String (Optional)	Name of the image as a string.
status	ImageStatus (Optional)	Value of the status of the image so that you can filter on "ACTIVE" for example.
marker	UUID (Optional)	UUID of the image at which you want to set a marker.
limit	Int (Optional)	Integer value for the limit of values to return.
type	String (Optional)	Value of the type of image, such as BASE, SERVER, or ALL. Possible values: BASE, SERVER, ALL. Default: ALL.

This operation does not accept a request body.

### 3.9.1.2. Response

#### Example 3.72. List images: JSON response

```
{
  "images": [
    {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "rel": "self",
          "href": "/v2/images/70a599e0-31e7-49b7-b260-868f441e862b"
        }
      ]
    }
  ]
}
```

```
        "href": "http://openstack.example.com/v2/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "self"
    },
    {
        "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "bookmark"
    },
    {
        "href": "http://glance.openstack.example.com/openstack/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/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
        "rel": "self"
    },
    {
        "href": "http://openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
        "rel": "bookmark"
    },
    {
        "href": "http://glance.openstack.example.com/openstack/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/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",
        "rel": "self"
    },
    {
        "href": "http://openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",
        "rel": "bookmark"
    },
    {
        "href": "http://glance.openstack.example.com/openstack/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/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/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/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/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/openstack/images/cedef40a-ed67-4d10-800e-17455edce175",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/cedef40a-ed67-4d10-800e-17455edce175",
                "rel": "bookmark"
            },
            {

```

```

        "href": "http://glance.openstack.example.com/openstack/
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/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
        "rel": "self"
},
{
        "href": "http://openstack.example.com/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
        "rel": "bookmark"
},
{
        "href": "http://glance.openstack.example.com/openstack/
images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
        "rel": "alternate",
        "type": "application/vnd.openstack.image"
}
],
"name": "fakeimage123456"
}
]
}

```

This table shows the body parameters for the list images response:

Name	Type	Description
images	Images WithOnly IDsNames Links <i>(Required)</i>	Image information.
next	AnyURI <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI <i>(Optional)</i>	Moves to the previous metadata item.

### Example 3.73. List images: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<images xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1">
    <image name="fakeimage7" id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://openstack.example.com/v2/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    
```

```
<atom:link href="http://glance.openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image name="fakeimage123456" id="155d900f-4e14-4e4c-a73d-069cbf4541e6">
    <atom:link href="http://openstack.example.com/v2/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image name="fakeimage123456" id="a2459075-d96c-40d5-893e-577ff92e721c">
    <atom:link href="http://openstack.example.com/v2/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image name="fakeimage6" id="a440c04b-79fa-479c-bed1-0b816eaec379">
    <atom:link href="http://openstack.example.com/v2/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image name="fakeimage123456" id="c905cedb-7281-47e4-8a62-f26bc5fc4c77">
    <atom:link href="http://openstack.example.com/v2/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image name="fakeimage123456" id="cedef40a-ed67-4d10-800e-17455edce175">
    <atom:link href="http://openstack.example.com/v2/openstack/images/cedef40a-ed67-4d10-800e-17455edce175" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/cedef40a-ed67-4d10-800e-17455edce175" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/cedef40a-ed67-4d10-800e-17455edce175" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image name="fakeimage123456" id="76fa36fc-c930-4bf3-8c8a-ea2a2420deb6">
    <atom:link href="http://openstack.example.com/v2/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" type="application/vnd.openstack.image" rel="alternate"/>
</image>
</images>
```

## 3.9.2. List images details

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/images/detail{?changes-since,server,name,status,marker,limit,type}	Lists all details for available images.

**Normal response codes:** 200, 203

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

### 3.9.2.1. Request

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

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

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

Name	Type	Description
changes-since	DateTime (Optional)	A time/date stamp for when the image last changed status.
server	AnyURI (Optional)	Name of the server in URL format.
name	String (Optional)	Name of the image as a string.
status	ImageStatus (Optional)	Value of the status of the image so that you can filter on "ACTIVE" for example.
marker	UUID (Optional)	UUID of the image at which you want to set a marker.
limit	Int (Optional)	Integer value for the limit of values to return.
type	String (Optional)	Value of the type of image, such as BASE, SERVER, or ALL. Possible values: BASE, SERVER, ALL. Default: ALL.

This operation does not accept a request body.

### 3.9.2.2. Response

#### Example 3.74. List images details: JSON response

```
{
  "images": [
    {
      "created": "2011-01-01T01:02:03Z",
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "rel": "self",
          "href": "/v2/images/70a599e0-31e7-49b7-b260-868f441e862b"
        }
      ]
    }
  ]
}
```

```
        "href": "http://openstack.example.com/v2/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "self"
    },
    {
        "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "bookmark"
    },
    {
        "href": "http://glance.openstack.example.com/openstack/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"
},
{
    "created": "2011-01-01T01:02:03Z",
    "id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
            "rel": "bookmark"
        },
        {
            "href": "http://glance.openstack.example.com/openstack/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/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/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/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/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/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/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/openstack/images/cedef40a-ed67-4d10-800e-17455edce175",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/cedef40a-ed67-4d10-800e-17455edce175",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/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/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/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 table shows the body parameters for the list images details response:

Name	Type	Description
next	AnyURI <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI <i>(Optional)</i>	Moves to the previous metadata item.

### Example 3.75. List images details: XML response

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

```
<images xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1">
    <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage7"
        created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
        "70a599e0-31e7-49b7-b260-868f441e862b">
        <metadata>
            <meta key="kernel_id">nokernel</meta>
            <meta key="auto_disk_config">True</meta>
            <meta key="ramdisk_id">nokernel</meta>
            <meta key="architecture">x86_64</meta>
        </metadata>
        <atom:link href="http://openstack.example.com/v2/openstack/images/
        70a599e0-31e7-49b7-b260-868f441e862b" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/images/
        70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        <atom:link href="http://glance.openstack.example.com/openstack/images/
        70a599e0-31e7-49b7-b260-868f441e862b" type="application/vnd.openstack.image"
        rel="alternate"/>
    </image>
    <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
        created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
        "155d900f-4e14-4e4c-a73d-069cbf4541e6">
        <metadata>
            <meta key="kernel_id">nokernel</meta>
            <meta key="ramdisk_id">nokernel</meta>
            <meta key="architecture">x86_64</meta>
        </metadata>
        <atom:link href="http://openstack.example.com/v2/openstack/images/
        155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/images/
        155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="bookmark"/>
        <atom:link href="http://glance.openstack.example.com/openstack/images/
        155d900f-4e14-4e4c-a73d-069cbf4541e6" type="application/vnd.openstack.image"
        rel="alternate"/>
    </image>
    <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
        created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
        "a2459075-d96c-40d5-893e-577ff92e721c">
        <metadata>
            <meta key="kernel_id">nokernel</meta>
            <meta key="ramdisk_id">nokernel</meta>
        </metadata>
        <atom:link href="http://openstack.example.com/v2/openstack/images/
        a2459075-d96c-40d5-893e-577ff92e721c" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/images/a2459075-
        d96c-40d5-893e-577ff92e721c" rel="bookmark"/>
        <atom:link href="http://glance.openstack.example.com/openstack/images/
        a2459075-d96c-40d5-893e-577ff92e721c" type="application/vnd.openstack.image"
        rel="alternate"/>
    </image>
    <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage6"
        created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
        "a440c04b-79fa-479c-bed1-0b816eaec379">
        <metadata>
            <meta key="kernel_id">nokernel</meta>
            <meta key="auto_disk_config">False</meta>
            <meta key="ramdisk_id">nokernel</meta>
            <meta key="architecture">x86_64</meta>
        </metadata>
```

```
<atom:link href="http://openstack.example.com/v2/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="c905cedb-7281-47e4-8a62-f26bc5fc4c77">
<metadata>
<meta key="kernel_id">155d900f-4e14-4e4c-a73d-069cbf4541e6</meta>
<meta key="ramdisk_id">None</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="cedef40a-ed67-4d10-800e-17455edce175">
<metadata>
<meta key="kernel_id">nokernel</meta>
<meta key="ramdisk_id">nokernel</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/cedef40a-ed67-4d10-800e-17455edce175" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/cedef40a-ed67-4d10-800e-17455edce175" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/cedef40a-ed67-4d10-800e-17455edce175" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="76fa36fc-c930-4bf3-8c8a-ea2a2420deb6">
<metadata>
<meta key="kernel_id">nokernel</meta>
<meta key="ramdisk_id">nokernel</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" type="application/vnd.openstack.image" rel="alternate"/>
</image>
</images>
```

### 3.9.3. Get image details

Method	URI	Description
GET	/v2/{tenant_id}/images/{image_id}	Gets details for a specified image.

**Normal response codes:** 200, 203

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

#### 3.9.3.1. Request

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

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

This operation does not accept a request body.

#### 3.9.3.2. Response

##### Example 3.76. Get image details: JSON response

```
{
  "image": {
    "created": "2011-01-01T01:02:03Z",
    "id": "70a599e0-31e7-49b7-b260-868f441e862b",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "bookmark"
      },
      {
        "href": "http://glance.openstack.example.com/openstack/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,
    "minMemory": 256
  }
}
```

```
        "minRam": 0,
        "name": "fakeimage7",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    }
}
```

### Example 3.77. Get image details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<image xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage7" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="70a599e0-31e7-49b7-b260-868f441e862b">
    <metadata>
        <meta key="kernel_id">nokernel</meta>
        <meta key="auto_disk_config">True</meta>
        <meta key="ramdisk_id">nokernel</meta>
        <meta key="architecture">x86_64</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" type="application/vnd.openstack.image" rel="alternate"/>
</image>
```

## 3.9.4. Delete image

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/images/{image_id}	Deletes a specified image.

**Normal response codes:** 204

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

### 3.9.4.1. Request

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

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

This operation does not accept a request body.

## 3.10. Image metadata

Show details for, set, update, and delete image metadata or metadata items.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/images/{image_id}/metadata	Shows metadata for a specified image.
<b>PUT</b>	/v2/{tenant_id}/images/{image_id}/metadata	Creates or replaces metadata for a specified image.
<b>POST</b>	/v2/{tenant_id}/images/{image_id}/metadata	Updates metadata items by key for a specified image.
<b>GET</b>	/v2/{tenant_id}/images/{image_id}/metadata/{key}	Shows details for a metadata item by key for a specified image.
<b>PUT</b>	/v2/{tenant_id}/images/{image_id}/metadata/{key}	Creates or updates a metadata item by key for a specified image.
<b>DELETE</b>	/v2/{tenant_id}/images/{image_id}/metadata/{key}	Deletes a metadata item by key for a specified image.

## 3.10.1. Show image metadata

Method	URI	Description
GET	/v2/{tenant_id}/images/{image_id}/metadata	Shows metadata for a specified image.

**Normal response codes:** 200, 203

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

### 3.10.1.1. Request

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

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

This operation does not accept a request body.

### 3.10.1.2. Response

#### Example 3.78. Show image metadata: JSON response

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

This table shows the body parameters for the show image metadata response:

Name	Type	Description
next	AnyURI <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI <i>(Optional)</i>	Moves to the previous metadata item.

#### Example 3.79. Show image metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
  <meta key="kernel_id">nokernel</meta>
  <meta key="auto_disk_config">True</meta>
  <meta key="ramdisk_id">nokernel</meta>
  <meta key="architecture">x86_64</meta>
</metadata>
```



## 3.10.2. Create or replace image metadata

Method	URI	Description
PUT	/v2/{tenant_id}/images/{image_id}/metadata	Creates or replaces metadata for a specified image.

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

If the number of metadata items exceeds the quota for metadata items, an overLimit (413) fault might be thrown.

**Normal response codes:** 200

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

### 3.10.2.1. Request

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

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

#### Example 3.80. Create or replace image metadata: JSON request

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

#### Example 3.81. Create or replace image metadata: XML request

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

<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
  <meta key="auto_disk_config">True</meta>
  <meta key="Label">Changed</meta>
</metadata>
```

### 3.10.2.2. Response

#### Example 3.82. Create or replace image metadata: JSON response

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

{}

This table shows the body parameters for the create or replace image metadata response:

Name	Type	Description
next	AnyURI <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI <i>(Optional)</i>	Moves to the previous metadata item.

### Example 3.83. Create or replace image metadata: XML response

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

### 3.10.3. Update image metadata items

Method	URI	Description
POST	/v2/{tenant_id}/images/{image_id}/metadata	Updates metadata items by key for a specified image.

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

An overLimit (413) fault might be thrown if the operation causes the quota for metadata items to be exceeded.

**Normal response codes:** 200

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

#### 3.10.3.1. Request

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

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

#### Example 3.84. Update image metadata items: JSON request

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

#### Example 3.85. Update image metadata items: XML request

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

#### 3.10.3.2. Response

#### Example 3.86. Update image metadata items: JSON response

```
{
  "metadata": {
    "Label": "UpdatedImage",
    "architecture": "x86_64",
    "auto_disk_config": "True",
    "kernel_id": "False",
```

```
        "ramdisk_id": "nokernel"
    }
}
```

This table shows the body parameters for the update image metadata items response:

Name	Type	Description
next	AnyURI <i>(Optional)</i>	Moves to the next metadata item.
previous	AnyURI <i>(Optional)</i>	Moves to the previous metadata item.

### Example 3.87. Update image metadata items: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
    <meta key="kernel_id">False</meta>
    <meta key="ramdisk_id">nokernel</meta>
    <meta key="Label">UpdatedImage</meta>
    <meta key="architecture">x86_64</meta>
    <meta key="auto_disk_config">True</meta>
</metadata>
```

## 3.10.4. Show image metadata item details

Method	URI	Description
GET	/v2/{tenant_id}/images/{image_id}/metadata/{key}	Shows details for a metadata item by key for a specified image.

**Normal response codes:** 200, 203

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

### 3.10.4.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{image_id}	UUID	The UUID for the image.
{key}	Image Metadata Key	A string. Maximum length is 255 characters.

This operation does not accept a request body.

### 3.10.4.2. Response

#### Example 3.88. Show image metadata item details: JSON response

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

#### Example 3.89. Show image metadata item details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
    <meta key="kernel_id">nokernel</meta>
    <meta key="auto_disk_config">True</meta>
    <meta key="ramdisk_id">nokernel</meta>
    <meta key="architecture">x86_64</meta>
</metadata>
```

### 3.10.5. Create or update image metadata item

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

An overLimit (413) fault might be thrown if the operation causes the quota for metadata items to be exceeded.

**Normal response codes:** 200

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

#### 3.10.5.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{image_id}	UUID	The UUID for the image.
{key}	Image Metadata Key	A string. Maximum length is 255 characters.

#### Example 3.90. Create or update image metadata item: JSON request

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

#### Example 3.91. Create or update image metadata item: XML request

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

<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
    <meta key="auto_disk_config">True</meta>
    <meta key="Label">Changed</meta>
</metadata>
```

#### 3.10.5.2. Response

#### Example 3.92. Create or update image metadata item: JSON response

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

```
    }  
}
```

### Example 3.93. Create or update image metadata item: XML response

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

## 3.10.6. Delete image metadata item

Method	URI	Description
DELETE	/v2/{tenant_id}/images/{image_id}/metadata/{key}	Deletes a metadata item by key for a specified image.

**Normal response codes:** 204

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), buildInProgress (409)

### 3.10.6.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The tenant ID in a multi-tenancy cloud.
{image_id}	UUID	The UUID for the image.
{key}	Image Metadata Key	A string. Maximum length is 255 characters.

This operation does not accept a request body.

## 4. Compute API v2 extensions (CURRENT)

Extensions add features, MIME types, actions, states, headers, parameters, and resources to the core Compute API without requiring a version change.



### Note

XML support in requests and responses has been deprecated for the Compute API v2.

Method	URI	Description
Server admin actions (action)		
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Pauses a server. Changes its status to PAUSED.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Unpauses a PAUSED server. Changes its status to ACTIVE.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Suspends a server. Changes its status to SUSPENDED.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Resumes a SUSPENDED server. Changes its status to ACTIVE.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Migrates a server to a host. The scheduler chooses the host.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Resets networking on a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Injects network information into a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Locks a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Unlocks a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Backs up a server instance.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Live-migrates a server to a new host without rebooting.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Resets the state of a server to a specified state.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Evacuates a server from failed host.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Assigns a specified security group to a specified server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Removes a specified security group from a specified server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to an instance.
Server diagnostics (diagnostics)		
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/diagnostics	Gets basic usage data for a specified server.
Flavor access (flavors)		
<b>GET</b>	/v2/{tenant_id}/flavors	Lists flavors and includes the access type, which is public or private.
<b>POST</b>	/v2/{tenant_id}/flavors	Creates a private flavor.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}	Gets the flavor access type, which is public or private.
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}/os-flavor-access	Lists tenants with access to the specified private flavor.
<b>POST</b>	/v2/{tenant_id}/flavors/{flavor_id}/action	Gives a specified tenant access to the specified private flavor.
<b>DELETE</b>	/v2/{tenant_id}/flavors/{flavor_id}/action	Revokes access from the specified tenant for the specified private flavor.
Flavors with FlavorDisabled attribute (flavors)		
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the OS-FLV-DISABLED:disabled attribute.
<b>GET</b>	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the OS-FLV-DISABLED:disabled attribute.
Flavor extra-specs (os-extra-specs)		
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Lists the extra-specs or keys for the specified flavor.
<b>POST</b>	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Creates extra-specs or keys for the specified flavor.
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{key_id}	Gets the value of the specified key.
<b>DELETE</b>	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{key_id}	Deletes a specified extra-spec by key.
Flavors with rtx_factor extended attribute (flavors)		
<b>POST</b>	/v2/{tenant_id}/flavors	Creates a flavor. Includes the rtx_factor extended attribute.
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the rtx_factor extended attribute.
<b>GET</b>	/v2/{tenant_id}/flavors/detail	Lists details for available flavors and includes the rtx_factor extended attribute.
Flavors with extended attributes (flavors)		
<b>POST</b>	/v2/{tenant_id}/flavors	Creates a flavor. Includes the rtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the rtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.
<b>GET</b>	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the rtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.
Flavors create or delete (flavors)		
<b>POST</b>	/v2/{tenant_id}/flavors	Creates a flavor.
<b>DELETE</b>	/v2/{tenant_id}/flavors/{flavor_id}	Deletes a flavor.
Images with size attribute (images)		
<b>GET</b>	/v2/{tenant_id}/images/detail	Lists details for available images. Includes the image size.
<b>GET</b>	/v2/{tenant_id}/images/{image_id}	Gets details for a specified image. Includes the image size.
Limits with project usage (limits)		
<b>GET</b>	/v2/{tenant_id}/limits	Gets absolute and rate limit information, including information on currently used absolute limits.
Limits with project usage for administrators (limits)		

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/limits/{tenant_id}	Enables administrators to get absolute and rate limit information, including information about currently used absolute limits, for a specified customer tenant ID.
Guest agents (os-agents)		
<b>GET</b>	/v2/{tenant_id}/os-agents	Lists all agent builds.
<b>POST</b>	/v2/{tenant_id}/os-agents	Creates an agent build.
<b>DELETE</b>	/v2/{tenant_id}/os-agents	Deletes an existing agent build.
<b>PUT</b>	/v2/{tenant_id}/os-agents/{id}	Updates an agent build.
Host aggregates (os-aggregates)		
<b>GET</b>	/v2/{tenant_id}/os-aggregates	Lists all aggregates.
<b>POST</b>	/v2/{tenant_id}/os-aggregates	Creates an aggregate.
<b>DELETE</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Deletes an aggregate.
<b>GET</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Gets details about a specified aggregate.
<b>PUT</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Updates the name, and optionally the availability zone, for a specified aggregate.
<b>POST</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Sets metadata for an aggregate.
<b>POST</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Adds a host to an aggregate.
<b>POST</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Removes a host from an aggregate.
Attach interfaces (os-interface)		
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/os-interface	Creates and uses a port interface to attach the port to a server instance.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-interface	Lists port interfaces.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}	Shows information about a specified port interface.
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}	Detaches a specified port interface.
Root certificates (os-certificates)		
<b>POST</b>	/v2/{tenant_id}/os-certificates	Creates a root certificate.
<b>GET</b>	/v2/{tenant_id}/os-certificates	Shows details for a root certificate owned by a specified tenant ID.
Cloudpipe (os-cloudpipe)		
<b>GET</b>	/v2/{tenant_id}/os-cloudpipe	Lists cloudpipes.
<b>POST</b>	/v2/{tenant_id}/os-cloudpipe{?project_id}	Creates a cloudpipe.
<b>POST</b>	/v2/{tenant_id}/os-cloudpipe/configure-project	Updates the virtual private network (VPN) IP address and port for a specified cloudpipe instance.
Server console output (os-console-output)		
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Gets console output for a server instance.
Server console (os-consoles)		
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Gets a console for a server instance.
Coverage reports (os-coverage)		

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/os-coverage/action	Generates a coverage report.
<b>POST</b>	/v2/{tenant_id}/os-coverage/action	Starts Nova coverage reporting.
<b>POST</b>	/v2/{tenant_id}/os-coverage/action	Starts coverage reporting for all services.
<b>POST</b>	/v2/{tenant_id}/os-coverage/action	Stops coverage reporting.
Server deferred delete (os-deferred-delete)		
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Force-deletes a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Restores a deleted server.
Fixed IPs (os-fixed-ips)		
<b>GET</b>	/v2/{tenant_id}/os-fixed-ips/{fixed_ip}	Shows information for a specified fixed IP address.
<b>POST</b>	/v2/{tenant_id}/os-fixed-ips/{fixed_ip}/action	Reserves or releases a fixed IP.
Floating IP DNS records (os-floating-ip-dns)		
<b>GET</b>	/v2/{tenant_id}/os-floating-ip-dns	Lists registered DNS domains published by the DNS drivers.
<b>PUT</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}	Creates or updates a DNS domain.
<b>DELETE</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}	Deletes a DNS domain and all associated host entries.
<b>PUT</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Creates or updates a DNS entry.
<b>GET</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Finds a unique DNS entry for a specified domain and name.
<b>DELETE</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Deletes a specified DNS entry.
<b>GET</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}	Lists DNS entries for a specified domain and IP.
Floating IP pools (os-floating-ip-pools)		
<b>GET</b>	/v2/{tenant_id}/os-floating-ip-pools	Lists floating IP pools.
Floating IPs (os-floating-ips)		
<b>GET</b>	/v2/{tenant_id}/os-floating-ips	Lists floating IP addresses associated with the tenant or account.
<b>POST</b>	/v2/{tenant_id}/os-floating-ips	Allocates a new floating IP address to a tenant or account.
<b>GET</b>	/v2/{tenant_id}/os-floating-ips/{id}	Shows information for a specified floating IP address.
<b>DELETE</b>	/v2/{tenant_id}/os-floating-ips/{id}	Deallocates the floating IP address associated with floating_IP_address_ID.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to an instance.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Removes a floating IP from an instance.
Floating IPs bulk (os-floating-ips-bulk)		
<b>GET</b>	/v2/{tenant_id}/os-floating-ips-bulk	Lists all floating IPs.
<b>POST</b>	/v2/{tenant_id}/os-floating-ips-bulk	Bulk-creates floating IPs.
<b>POST</b>	/v2/{tenant_id}/os-floating-ips-bulk/delete	Bulk-deletes floating IPs.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-floating-ips-bulk/{host_name}	Lists all floating IPs for a specified host.
Hosts (os-hosts)		
<b>GET</b>	/v2/{tenant_id}/os-hosts{?service, zone}	Lists hosts.
<b>GET</b>	/v2/{tenant_id}/os-hosts/{host_name}	Shows information for a specified host.
<b>PUT</b>	/v2/{tenant_id}/os-hosts/{host_name}	Enables a host or puts it in maintenance mode.
<b>GET</b>	/v2/{tenant_id}/os-hosts/{host_name}/startup	Starts a host.
<b>GET</b>	/v2/{tenant_id}/os-hosts/{host_name}/shutdown	Shuts down a host.
<b>GET</b>	/v2/{tenant_id}/os-hosts/{host_name}/reboot	Reboots a host.
Hypervisors (os-hypervisors)		
<b>GET</b>	/v2/{tenant_id}/os-hypervisors	Lists hypervisors information for each server obtained through the hypervisor-specific API, such as libvirt or XenAPI.
<b>GET</b>	/v2/{tenant_id}/os-hypervisors/detail	Shows information for a specified hypervisor. Typically configured as an admin-only extension by using policy.json settings.
<b>GET</b>	/v2/{tenant_id}/os-hypervisors/statistics	Shows hypervisor statistics over all compute nodes.
<b>GET</b>	/v2/{tenant_id}/os-hypervisors/{hypervisor_hostname}	Shows the up time for a specified hypervisor.
<b>GET</b>	/v2/{tenant_id}/os-hypervisors/{hypervisor_hostname}/servers	Lists instances that belong to specific hypervisors.
Server actions (os-instance-actions)		
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-instance-actions	Lists available actions for a specified server. Deployers set permissions for this request in the policy.json file. By default, all users can list actions.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-instance-actions/{action_id}	Gets details for a specified action for a specified server instance. Deployers set permissions for this request in the policy.json file. By default, only administrators can get details for an action.
Keypairs (os-keypairs)		
<b>GET</b>	/v2/{tenant_id}/os-keypairs	Lists keypairs that are associated with the account.
<b>POST</b>	/v2/{tenant_id}/os-keypairs	Generates or imports a keypair.
<b>DELETE</b>	/v2/{tenant_id}/os-keypairs/{keypair_name}	Deletes a keypair.
<b>GET</b>	/v2/{tenant_id}/os-keypairs/{keypair_name}	Shows a keypair associated with the account.
Migrations (os-migrations)		
<b>GET</b>	/v2/{tenant_id}/os-migrations{?host,status,cell_name}	Enables an administrative user to fetch in-progress migrations for a region or specified cell in a region.
Networks (os-networks)		
<b>POST</b>	/v2/{tenant_id}	Creates a network.
<b>GET</b>	/v2/{tenant_id}/os-networks	Lists networks that are available to the tenant.
<b>POST</b>	/v2/{tenant_id}/os-networks/add	Adds a specified network to a project.
<b>GET</b>	/v2/{tenant_id}/os-networks/{id}	Shows information for a specified network.

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/os-networks/{id}	Deletes a specified network.
<b>POST</b>	/v2/{tenant_id}/os-networks/{id}/action	Associates a specified network with a host.
<b>POST</b>	/v2/{tenant_id}/os-networks/{id}/action	Disassociates the host from a specified network.
<b>POST</b>	/v2/{tenant_id}/os-networks/{id}/action	Disassociates a specified network from a project so that the network can be reused.
<b>POST</b>	/v2/{tenant_id}/os-networks/{id}/action	Disassociates the project from a specified network.
Quota sets (os-quota-sets)		
<b>GET</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Shows quotas for a tenant.
<b>PUT</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.
<b>DELETE</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Deletes quotas for a tenant so the quotas revert to default values.
<b>GET</b>	/v2/{tenant_id}/os-quota-sets/defaults	Gets default quotas for a tenant.
<b>GET</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a specified tenant and user.
<b>POST</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a specified tenant/project and user.
<b>DELETE</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Deletes quotas for a user so that the quotas revert to default values.
<b>GET</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}	Shows details for quotas for a specified tenant and user.
Rules for default security group (os-security-group-default-rules)		
<b>GET</b>	/v2/{tenant_id}/os-security-group-default-rules	Lists default security group rules.
<b>POST</b>	/v2/{tenant_id}/os-security-group-default-rules	Creates a default security group rule.
<b>GET</b>	/v2/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Shows information for a specified security group rule.
<b>DELETE</b>	/v2/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Deletes a specified security group rule.
Security groups (os-security-groups)		
<b>GET</b>	/v2/{tenant_id}/os-security-groups	Lists security groups.
<b>POST</b>	/v2/{tenant_id}/os-security-groups	Creates a security group.
<b>GET</b>	/v2/{tenant_id}/os-security-groups/servers/{server_id}/os-security-groups	Lists security groups for a specified server.
<b>GET</b>	/v2/{tenant_id}/os-security-groups/{security_group_id}	Shows information for a specified security group.
<b>DELETE</b>	/v2/{tenant_id}/os-security-groups/{security_group_id}	Deletes a specified security group.
Server groups (os-server-groups)		
<b>GET</b>	/v2/{tenant_id}/os-server-groups	Lists server groups.
<b>POST</b>	/v2/{tenant_id}/os-server-groups	Creates a server group.
<b>GET</b>	/v2/{tenant_id}/os-server-groups/{ServerGroup_id}	Shows details for a specified server group.

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/os-server-groups/{ServerGroup_id}	Deletes a specified server group.
Server password (os-server-password)		
<b>GET</b>	/v2/servers/{server_id}/os-server-password	Gets the administrative password for a specified server.
<b>DELETE</b>	/v2/servers/{server_id}/os-server-password	Clears the encrypted copy of the password in the metadata server. This is done after the client has retrieved the password and knows it doesn't need it in the metadata server anymore. The password for the server remains the same.
Server shelfe (servers)		
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Shelves a running server and changes its status to SHELVED_OFFLOADED.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Removes an instance from the server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Restores a shelved server.
Server start and stop (servers)		
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Starts a stopped server and changes its status to ACTIVE.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Stops a running server and changes its status to STOPPED.
Manage services (os-services)		
<b>GET</b>	/v2/{tenant_id}/os-services	Lists running services.
<b>PUT</b>	/v2/{tenant_id}/os-services/enable	Enables scheduling for a service.
<b>PUT</b>	/v2/{tenant_id}/os-services/disable	Disables scheduling for a service.
<b>PUT</b>	/v2/{tenant_id}/os-services/disable-log-reason	Logs information to the service table about why a service was disabled.
<b>GET</b>	/v2/{tenant_id}/os-services/detail	Lists disabled services. If information exists, includes reasons why services were disabled.
Usage reports (os-simple-tenant-usage)		
<b>GET</b>	/v2/{tenant_id}/os-simple-tenant-usage	Lists usage information for all tenants.
<b>GET</b>	/v2/{tenant_id}/os-simple-tenant-usage/{tenant_id}	Gets usage information for a tenant.
Virtual interfaces (os-virtual-interfaces)		
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Lists the virtual interfaces for a specified instance.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Shows the virtual interface for a specified instance. Includes the OS-EXT-VIF-NET:net_id attribute that shows to which network the interface is attached.
Volume extension (os-volumes, os-snapshots)		
<b>GET</b>	/v1.1/{tenant_id}/os-volumes	Lists the volumes associated with the account.
<b>GET</b>	/v1.1/{tenant_id}/os-volumes/detail	Lists details for a specified volume.
<b>POST</b>	/v1.1/{tenant_id}/os-volumes/{volume_id}	Creates a volume.
<b>GET</b>	/v1.1/{tenant_id}/os-volumes/{volume_id}	Shows information for a specified volume.
<b>DELETE</b>	/v1.1/{tenant_id}/os-volumes/{volume_id}	Deletes a specified volume.

Method	URI	Description
<b>GET</b>	/v1.1/{tenant_id}/os-volume-types	Lists volume types.
<b>GET</b>	/v1.1/{tenant_id}/os-volume-types/{volume_type_id}	Shows information for a specified volume type.
<b>POST</b>	/v1.1/{tenant_id}/os-snapshots	Creates a snapshot.
<b>GET</b>	/v1.1/{tenant_id}/os-snapshots	Lists snapshots.
<b>GET</b>	/v1.1/{tenant_id}/os-snapshots/detail	Lists details for a specified snapshot.
<b>GET</b>	/v1.1/{tenant_id}/os-snapshots/{snapshot_id}	Shows information for a specified snapshot.
<b>DELETE</b>	/v1.1/{tenant_id}/os-snapshots/{snapshot_id}	Deletes a specified snapshot from the account.
Volume attachments (os-volume_attachments)		
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Attaches a volume to the specified server.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Lists the volume attachments for a specified server.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Shows details for the specified volume attachment.
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Deletes the specified volume attachment from a specified server.
Servers with block device mapping format (servers)		
<b>GET</b>	/v2/{tenant_id}/servers{?changes-since,image,flavor,name,marker,limit,status,host}	Lists IDs, names, and links for all servers.
<b>POST</b>	/v2/{tenant_id}/servers	Creates a server with a block device mapping.
Server OS-EXT-IPS-MAC:mac_addr extended attribute (servers)		
<b>POST</b>	/v2/{tenant_id}/servers	Creates a server with the OS-EXT-IPS-MAC:mac_addr extended attribute.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}{?changes-since,image,flavor,name,marker,limit,status,host}	Shows information for a specified server. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.
<b>GET</b>	/v2/{tenant_id}/servers/detail{?changes-since,image,flavor,name,marker,limit,status,host}	Lists details for all servers. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.
Configuration drive (servers)		
<b>POST</b>	/v2/{tenant_id}/servers	Creates a server with the configuration drive extended attribute.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server including the configuration drive extended attribute.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/detail	Lists details for all servers including the configuration drive extended attribute.
Servers with extended availability zones (servers)		
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server, including its availability zone.
<b>GET</b>	/v2/{tenant_id}/servers/detail	Lists details for servers, including their current availability zone.
Servers and images with disk config (servers, images)		
<b>POST</b>	/v2/{tenant_id}/servers	Creates a server.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server.
<b>PUT</b>	/v2/{tenant_id}/servers/{server_id}	Updates a specified server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Resizes a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Rebuilds a specified server.
<b>GET</b>	/v2/{tenant_id}/servers/detail	Lists servers.
<b>GET</b>	/v2/{tenant_id}/images/{image_id}	Gets information for a specified image.
<b>GET</b>	/v2/{tenant_id}/images/detail	Lists images.
Server IP type (servers)		
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/action	Shows the type of IP assigned to a specified server, either fixed or floating.
<b>GET</b>	/v2/{tenant_id}/servers/detail	Lists servers and shows their IPs by type. Type is either fixed or floating.
Server extended attributes (servers)		
<b>GET</b>	/v2/servers	Lists detailed extended server attribute information for all servers.
<b>GET</b>	/v2/servers/{server_id}	Shows extended server attributes for a specified server.
Server extended status (servers)		
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}	Shows the extended status attributes in the response for a specified server.
<b>GET</b>	/v2/{tenant_id}/servers/detail	Lists the extended status attributes in the detailed response for all servers.
Servers multiple create (servers)		
<b>POST</b>	/v2/{tenant_id}/servers{?return_reservation_id,min_count,max_count}	Creates one or more servers with an optional reservation ID.
Server rescue and unrescue (servers)		
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Puts a server in rescue mode. Changes status to RESCUE.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Unrescues a server.
Servers with scheduler hints (servers)		
<b>POST</b>	/v2/{tenant_id}/servers	Creates a server with scheduler hints that are passed directly to the scheduler.

## 4.1. Server admin actions (action)

Administrator only. Perform actions on a server. Specify the action in the request body.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Pauses a server. Changes its status to PAUSED.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Unpauses a PAUSED server. Changes its status to ACTIVE.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Suspends a server. Changes its status to SUSPENDED.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Resumes a SUSPENDED server. Changes its status to ACTIVE.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Migrates a server to a host. The scheduler chooses the host.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Resets networking on a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Injects network information into a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Locks a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Unlocks a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Backs up a server instance.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Live-migrates a server to a new host without rebooting.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Resets the state of a server to a specified state.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Evacuates a server from failed host.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Assigns a specified security group to a specified server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Removes a specified security group from a specified server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to an instance.

## 4.1.1. Pause server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Pauses a server. Changes its status to PAUSED.

This operation stores the state of the VM in RAM. A paused instance continues to run in a frozen state.

**Normal response codes:** 202

### 4.1.1.1. Request

This table shows the URI parameters for the pause server 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 4.1. Pause server: JSON request

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

This table shows the body parameters for the pause server request:

Name	Type	Description
pause	String <i>(Required)</i>	Specify the pause action in the request body.

#### Example 4.2. Pause server: XML request

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

This operation does not accept a request body.

## 4.1.2. Unpause server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Unpauses a PAUSED server. Changes its status to ACTIVE.

Normal response codes: 202

### 4.1.2.1. Request

This table shows the URI parameters for the unpause server 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 4.3. Unpause server: JSON request

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

This table shows the body parameters for the unpause server request:

Name	Type	Description
unpause	String <i>(Required)</i>	Specify the unpause action in the request body.

#### Example 4.4. Unpause server: XML request

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

This operation does not accept a request body.

## 4.1.3. Suspend server

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

This operation stores the VM state on disk, writes all memory to disk, and stops the virtual machine.

Administrative users might suspend an infrequently used instance or suspend an instance to perform system maintenance. Suspending an instance is similar to placing a device in hibernation; memory and vCPUs become available to create other instances.

**Normal response codes:** 202

### 4.1.3.1. Request

This table shows the URI parameters for the suspend server 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 4.5. Suspend server: JSON request

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

This table shows the body parameters for the suspend server request:

Name	Type	Description
suspend	String <i>(Required)</i>	Specify the suspend action in the request body.

#### Example 4.6. Suspend server: XML request

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

This operation does not accept a request body.

## 4.1.4. Resume server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Resumes a SUSPENDED server. Changes its status to ACTIVE.

Normal response codes: 202

### 4.1.4.1. Request

This table shows the URI parameters for the resume server 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 4.7. Resume server: JSON request

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

This table shows the body parameters for the resume server request:

Name	Type	Description
resume	String <i>(Required)</i>	Specify the resume action in the request body.

#### Example 4.8. Resume server: XML request

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

This operation does not accept a request body.

## 4.1.5. Migrate server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Migrates a server to a host. The scheduler chooses the host.

Normal response codes: 202

### 4.1.5.1. Request

This table shows the URI parameters for the migrate server 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 4.9. Migrate server: JSON request

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

This table shows the body parameters for the migrate server request:

Name	Type	Description
migrate	String <i>(Required)</i>	Specify the <code>migrate</code> action in the request body.

#### Example 4.10. Migrate server: xml request

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

This operation does not accept a request body.

## 4.1.6. Reset networking on server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Resets networking on a server.

Normal response codes: 202

### 4.1.6.1. Request

This table shows the URI parameters for the reset networking on server 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 4.11. Reset network: JSON request

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

This table shows the body parameters for the reset networking on server request:

Name	Type	Description
resetNetwork	String <i>(Required)</i>	Specify the <code>resetNetwork</code> action in the request body.

#### Example 4.12. Reset network: XML request

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

This operation does not accept a request body.

## 4.1.7. Inject network information

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Injects network information into a server.

Normal response codes: 202

### 4.1.7.1. Request

This table shows the URI parameters for the inject network information 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 4.13. Insert network information: JSON request

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

This table shows the body parameters for the inject network information request:

Name	Type	Description
injectNetworkInfo	String <i>(Required)</i>	Specify the injectNetworkInfo action in the request body.

#### Example 4.14. Insert network information: XML request

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

This operation does not accept a request body.

## 4.1.8. Lock server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Locks a server.

Normal response codes: 202

### 4.1.8.1. Request

This table shows the URI parameters for the lock server 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 4.15. Lock server: JSON request

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

This table shows the body parameters for the lock server request:

Name	Type	Description
lock	String <i>(Required)</i>	Specify the lock action in the request body.

#### Example 4.16. Lock server: XML request

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

This operation does not accept a request body.

## 4.1.9. Unlock server

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

Normal response codes: 202

### 4.1.9.1. Request

This table shows the URI parameters for the unlock server 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 4.17. Unlock server: JSON request

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

This table shows the body parameters for the unlock server request:

Name	Type	Description
unlock	String <i>(Required)</i>	Specify the unlock action in the request body.

#### Example 4.18. Unlock server: XML request

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

This operation does not accept a request body.

## 4.1.10. Create server backup

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Backs up a server instance.

Normal response codes: 202

### 4.1.10.1. Request

This table shows the URI parameters for the create server backup 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 4.19. Create server backup: JSON request

```
{
  "createBackup": {
    "name": "Backup_1",
    "backup_type": "daily",
    "rotation": 1
  }
}
```

This table shows the body parameters for the create server backup request:

Name	Type	Description
createBackup	String <i>(Required)</i>	Specify the createBackup action in the request body.
name	String <i>(Required)</i>	Name of the backup image.
backup_type	String <i>(Required)</i>	The backup type. Either 'daily' or 'weekly.'
rotation	Int <i>(Required)</i>	An integer value that represents the number of backups to maintain.

#### Example 4.20. Create server backup: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<createBackup>
  <name>Backup_1</name>
  <backup_type>daily</backup_type>
  <rotation>1</rotation>
</createBackup>
```

This operation does not accept a request body.

## 4.1.11. Live-migrate server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Live-migrates a server to a new host without rebooting.

Normal response codes: 202

### 4.1.11.1. Request

This table shows the URI parameters for the live-migrate server 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 4.21. Live-migrate server: JSON request

```
{
  "os-migrateLive": {
    "host": "0443e9a1254044d8b99f35eace132080",
    "block_migration": false,
    "disk_over_commit": false
  }
}
```

This table shows the body parameters for the live-migrate server request:

Name	Type	Description
os-migrateLive	String <i>(Required)</i>	Specify the os-migrateLive action in the request body.
host	String <i>(Required)</i>	Name of the new host.
block_migration	Boolean <i>(Required)</i>	If true, migration is blocked.
disk_over_commit	Boolean <i>(Required)</i>	If true, disk over commit is allowed.

#### Example 4.22. Live-migrate server: XML request

```
<?xml version="1.0" encoding="UTF-8" ?>
<os-migrateLive>
  <host>6217bf142bb3491995f966af999f9f23</host>
  <block_migration>false</block_migration>
  <disk_over_commit>false</disk_over_commit>
</os-migrateLive>
```

This operation does not accept a request body.

## 4.1.12. Reset server state

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Resets the state of a server to a specified state.

Normal response codes: 202

### 4.1.12.1. Request

This table shows the URI parameters for the reset server state 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 4.23. Reset server state: JSON request

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

This table shows the body parameters for the reset server state request:

Name	Type	Description
resetState	String <i>(Required)</i>	Specify the resetState action in the request body.
state	String <i>(Required)</i>	The desired state of a server.

#### Example 4.24. Reset server state: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<os-resetState>
  <state>active</state>
</os-resetState>
```

This operation does not accept a request body.

## 4.1.13. Evacuate server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Evacuates a server from failed host.

**Normal response codes:** 200

### 4.1.13.1. Request

This table shows the URI parameters for the evacuate server 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 4.25. Evacuate server: JSON request

```
{
  "evacuate": {
    "host": "TargetHost",
    "adminPass": "MySecretPass",
    "onSharedStorage": "True"
  }
}
```

This table shows the body parameters for the evacuate server request:

Name	Type	Description
evacuate	String <i>(Required)</i>	Specify the evacuate action in the request body.
host	String <i>(Required)</i>	The name or ID of the host where the server is evacuated.
adminPass	String <i>(Optional)</i>	Specifies a new admin password for the evacuated instance.
onSharedStorage	String <i>(Required)</i>	If server is on shared storage, set to True. Otherwise, set to False.

#### Example 4.26. Evacuate server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<evacuate xmlns="http://docs.openstack.org/compute/api/v2"
  host="TargetHost"
  adminPass="MySecretPass"
  onSharedStorage="True" />
```

This operation does not accept a request body.

### 4.1.13.2. Response

#### Example 4.27. Evacuate server: JSON response

```
{
```

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

### Example 4.28. Evacuate server: XML response

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

This operation does not return a response body.

## 4.1.14. Add security group

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Assigns a specified security group to a specified server.

Normal response codes: 200

### 4.1.14.1. Request

This table shows the URI parameters for the add security group 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 4.29. Add security group: JSON request

```
{
  "addSecurityGroup": {
    "name": "test"
  }
}
```

This table shows the body parameters for the add security group request:

Name	Type	Description
addSecurityGroup	String <i>(Required)</i>	Specify the addSecurityGroup action in the request body.
name	String <i>(Required)</i>	The name of the security group to assign to the server.

#### Example 4.30. Add security group: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<addSecurityGroup>
  <name>test</name>
</addSecurityGroup>
```

This operation does not accept a request body.

## 4.1.15. Remove security group

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Removes a specified security group from a specified server.

Normal response codes: 200

### 4.1.15.1. Request

This table shows the URI parameters for the remove security group 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 4.31. Remove security group: JSON request

```
{
  "removeSecurityGroup": {
    "name": "test"
  }
}
```

This table shows the body parameters for the remove security group request:

Name	Type	Description
removeSecurityGroup	String <i>(Required)</i>	Specify the removeSecurityGroup action in the request body.
name	String <i>(Required)</i>	The name of the security group to remove from the server.

#### Example 4.32. Remove security group: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<removeSecurityGroup>
  <name>test</name>
</removeSecurityGroup>
```

This operation does not accept a request body.

## 4.1.16. Add floating IP address

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to an instance.

You can optionally associate a fixed IP address with the floating IP address.

**Normal response codes:** 200

### 4.1.16.1. Request

This table shows the URI parameters for the add floating ip address 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 4.33. Add floating IP address: JSON request

```
{
  "addFloatingIp": {
    "fixed_address": "166.78.185.201",
    "address": "172.24.4.225"
  }
}
```

## 4.2. Server diagnostics (diagnostics)

Get the usage data for a server.

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

## 4.2.1. Get server diagnostics

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

Normal response codes: 200

### 4.2.1.1. Request

This table shows the URI parameters for the get server diagnostics 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.

### 4.2.1.2. Response

#### Example 4.34. Server diagnostics: JSON response

```
{
    "vnet0_tx_errors": 0,
    "vda_errors": -1,
    "vda_read": 4447232,
    "vda_write": 4347904,
    "vnet0_tx_packets": 1259,
    "vda_write_req": 3523,
    "memory-actual": 524288,
    "cpu0_time": 195230000000,
    "vnet0_tx": 364840,
    "vnet0_rx_drop": 0,
    "vnet0_rx_packets": 1423,
    "vnet0_rx_errors": 0,
    "memory": 524288,
    "memory-rss": 243188,
    "vda_read_req": 291,
    "vnet0_rx": 363725,
    "vnet0_tx_drop": 0
}
```

## 4.3. Flavor access (flavors)

Create and get details for private flavors. Also, list, add, and remove tenants' access to private flavors.

Method	URI	Description
GET	/v2/{tenant_id}/flavors	Lists flavors and includes the access type, which is public or private.
POST	/v2/{tenant_id}/flavors	Creates a private flavor.
GET	/v2/{tenant_id}/flavors/{flavor_id}	Gets the flavor access type, which is public or private.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}/os-flavor-access	Lists tenants with access to the specified private flavor.
<b>POST</b>	/v2/{tenant_id}/flavors/{flavor_id}/action	Gives a specified tenant access to the specified private flavor.
<b>DELETE</b>	/v2/{tenant_id}/flavors/{flavor_id}/action	Revokes access from the specified tenant for the specified private flavor.

## 4.3.1. List flavors with access type

Method	URI	Description
GET	/v2/{tenant_id}/flavors	Lists flavors and includes the access type, which is public or private.

**Normal response codes:** 200

### 4.3.1.1. Request

This table shows the URI parameters for the list flavors with access type 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.

### 4.3.1.2. Response

#### Example 4.35. List flavors with access type: JSON response

```
{
  "flavors": [
    {
      "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,
      "vcpus": 1
    },
    {
      "disk": 20,
      "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",
        "os-flavor-access:is_public": true,
        "ram": 2048,
        "vcpus": 1
    },
    {
        "disk": 40,
        "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",
        "os-flavor-access:is_public": true,
        "ram": 4096,
        "vcpus": 2
    },
    {
        "disk": 80,
        "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",
        "os-flavor-access:is_public": true,
        "ram": 8192,
        "vcpus": 4
    },
    {
        "disk": 160,
        "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",
        "os-flavor-access:is_public": true,
        "ram": 16384,
        "vcpus": 8
    }
]
}

```

This table shows the body parameters for the list flavors with access type response:

Name	Type	Description
disk	Int <i>(Required)</i>	Integer value for the minimum disk space in GB.
id	Int <i>(Required)</i>	The flavor ID.
links	String <i>(Required)</i>	Flavor links.
name	String <i>(Required)</i>	The flavor name.
ram	Int <i>(Required)</i>	Amount of RAM.
vcpus	Int <i>(Required)</i>	Number of VCPUs.
os-flavor-access:is_public	Boolean <i>(Required)</i>	Set to true if access is public.

### Example 4.36. List flavors with access type: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:os-flavor-access="http://docs.openstack.org/compute/ext/
flavor_access/api/v2" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/compute/api/v1.1">
    <flavor disk="1" vcpus="1" ram="512" name="m1.tiny" id="1" os-flavor-
access:is_public="True">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <flavor disk="20" vcpus="1" ram="2048" name="m1.small" id="2" os-flavor-
access:is_public="True">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/2" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/2" rel=
"bookmark"/>
    </flavor>
    <flavor disk="40" vcpus="2" ram="4096" name="m1.medium" id="3" os-flavor-
access:is_public="True">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/3" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/3" rel=
"bookmark"/>
    </flavor>

```

```
</flavor>
<flavor disk="80" vcpus="4" ram="8192" name="m1.large" id="4" os-flavor-
access:is_public="True">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/4" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/4" rel=
"bookmark"/>
</flavor>
<flavor disk="160" vcpus="8" ram="16384" name="m1.xlarge" id="5" os-flavor-
access:is_public="True">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/5" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/5" rel=
"bookmark"/>
</flavor>
</flavors>
```

This operation does not return a response body.

## 4.3.2. Create private flavor

Method	URI	Description
POST	/v2/{tenant_id}/flavors	Creates a private flavor.

**Normal response codes:** 200

### 4.3.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.37. Create private flavor: JSON request

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

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

Name	Type	Description
disk	Int <i>(Required)</i>	Integer value for the minimum disk space in GB.
id	Int <i>(Required)</i>	The flavor ID.
name	String <i>(Required)</i>	The flavor name.
ram	Int <i>(Required)</i>	Amount of RAM.
vcpus	Int <i>(Required)</i>	Number of VCPUs.
os-flavor-access:is_public	Boolean <i>(Required)</i>	Set to true if access is public. Set to false if access is private.

#### Example 4.38. Create private flavor: XML request

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

```

name="test_flavor"
ram="1024"
vcpus="2"
disk="10"
id="10"
os-flavor-access:is_public="False"
/>

```

This operation does not accept a request body.

### 4.3.2.2. Response

#### Example 4.39. Create private flavor: JSON response

```
{
  "flavor": {
    "disk": 10,
    "id": "10",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/flavors/10",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/10",
        "rel": "bookmark"
      }
    ],
    "name": "test_flavor",
    "os-flavor-access:is_public": false,
    "ram": 1024,
    "vcpus": 2
  }
}
```

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

Name	Type	Description
disk	Int <i>(Required)</i>	Integer value for the minimum disk space in GB.
id	Int <i>(Required)</i>	The flavor ID.
links	String <i>(Required)</i>	Flavor links.
name	String <i>(Required)</i>	The flavor name.
ram	Int <i>(Required)</i>	Amount of RAM.
vcpus	Int <i>(Required)</i>	Number of VCPUs.
os-flavor-access:is_public	Boolean	Set to true if access is public.

Name	Type	Description
	(Required)	

#### Example 4.40. Create private flavor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:os-flavor-access="http://docs.openstack.org/compute/ext/
flavor_access/api/v2" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
/docs.openstack.org/compute/api/v1.1" disk="10" vcpus="2" ram="1024" name=
"test_flavor" id="10" os-flavor-access:is_public="False">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/10" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/10" rel=
"bookmark" />
</flavor>
```

This operation does not return a response body.

### 4.3.3. Show flavor access type

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}	Gets the flavor access type, which is public or private.

Normal response codes: 200

#### 4.3.3.1. Request

This table shows the URI parameters for the show flavor access type request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{flavor_id}	Uuid	The ID of the flavor of interest to you.

This operation does not accept a request body.

#### 4.3.3.2. Response

##### Example 4.41. Show flavor access type: JSON response

```
{
    "flavor": {
        "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,
        "vcpus": 1
    }
}
```

This table shows the body parameters for the show flavor access type response:

Name	Type	Description
disk	Int <i>(Required)</i>	Integer value for the minimum disk space in GB.
id	Int <i>(Required)</i>	The flavor ID.
links	String <i>(Required)</i>	Flavor links.

Name	Type	Description
name	String <i>(Required)</i>	The flavor name.
ram	Int <i>(Required)</i>	Amount of RAM.
vcpus	Int <i>(Required)</i>	Number of VCPUs.
os-flavor-access:is_public	Boolean <i>(Required)</i>	Set to true if access is public.

#### Example 4.42. Show flavor access type: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:os-flavor-access="http://docs.openstack.org/compute/ext/
flavor_access/api/v2" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
/docs.openstack.org/compute/api/v1.1" disk="1" vcpus="1" ram="512" name="m1.
tiny" id="1" os-flavor-access:is_public="True">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
</flavor>
```

This operation does not return a response body.

## 4.3.4. List tenants with access to private flavor

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}/os-flavor-access	Lists tenants with access to the specified private flavor.

Normal response codes: 200

### 4.3.4.1. Request

This table shows the URI parameters for the list tenants with access to private flavor request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{flavor_id}	Uuid	The ID of the flavor of interest to you.

This operation does not accept a request body.

### 4.3.4.2. Response

#### Example 4.43. List tenants with access to private flavor: JSON response

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

This table shows the body parameters for the list tenants with access to private flavor response:

Name	Type	Description
flavor_id	Int <i>(Required)</i>	The flavor ID.
tenant_id	String <i>(Required)</i>	The tenant ID.

#### Example 4.44. List tenants with access to private flavor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor_access>
  <access tenant_id="fake_tenant" flavor_id="10"/>
  <access tenant_id="openstack" flavor_id="10"/>
</flavor_access>
```

This operation does not return a response body.

## 4.3.5. Add access to private flavor

Method	URI	Description
POST	/v2/{tenant_id}/flavors/{flavor_id}/action	Gives a specified tenant access to the specified private flavor.

Normal response codes: 200

### 4.3.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{flavor_id}	Uuid	The ID of the flavor of interest to you.

#### Example 4.45. Add access to private flavor: JSON request

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

This table shows the body parameters for the add access to private flavor request:

Name	Type	Description
addTenantAccess	String <i>(Required)</i>	Specify the addTenantAccess action.
tenant	String <i>(Required)</i>	The name of the tenant to which to give access.

#### Example 4.46. Add access to private flavor: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<addTenantAccess>
  <tenant>fake_tenant</tenant>
</addTenantAccess>
```

This operation does not accept a request body.

### 4.3.5.2. Response

#### Example 4.47. Add access to private flavor: JSON response

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

```
        "flavor_id": "10",
        "tenant_id": "openstack"
    }
]
```

This table shows the body parameters for the add access to private flavor response:

Name	Type	Description
flavor_id	Int <i>(Required)</i>	The flavor ID.
tenant_id	String <i>(Required)</i>	The tenant ID.

#### Example 4.48. Add access to private flavor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor_access>
    <access tenant_id="fake_tenant" flavor_id="10"/>
    <access tenant_id="openstack" flavor_id="10"/>
</flavor_access>
```

This operation does not return a response body.

## 4.3.6. Delete access from private flavor

Method	URI	Description
DELETE	/v2/{tenant_id}/flavors/{flavor_id}/action	Revokes access from the specified tenant for the specified private flavor.

Normal response codes: 200

### 4.3.6.1. Request

This table shows the URI parameters for the delete access from private flavor request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{flavor_id}	Uuid	The ID of the flavor of interest to you.

#### Example 4.49. Delete access from private flavor: JSON request

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

This table shows the body parameters for the delete access from private flavor request:

Name	Type	Description
removeTenantAccess	String <i>(Required)</i>	Specify the removeTenantAccess action.
tenant	String <i>(Required)</i>	The name of the tenant from which to revoke access.

#### Example 4.50. Delete access from private flavor: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<removeTenantAccess>
  <tenant>fake_tenant</tenant>
</removeTenantAccess>
```

This operation does not accept a request body.

### 4.3.6.2. Response

#### Example 4.51. Delete access from private flavor: JSON response

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

}

This table shows the body parameters for the delete access from private flavor response:

Name	Type	Description
flavor_id	Int <i>(Required)</i>	The flavor ID.
tenant_id	String <i>(Required)</i>	The tenant ID.

#### Example 4.52. Delete access from private flavor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor_access>
    <access tenant_id="openstack" flavor_id="10"/>
</flavor_access>
```

This operation does not return a response body.

## 4.4. Flavors with FlavorDisabled attribute (flavors)

Get details for a flavor, and list details for available flavors. Includes the OS-FLV-DISABLED:disabled extended attribute.

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the OS-FLV-DISABLED:disabled attribute.
GET	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the OS-FLV-DISABLED:disabled attribute.

## 4.4.1. Get flavor disabled status details

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the OS-FLV-DISABLED:disabled attribute.

Normal response codes: 200

### 4.4.1.1. Request

This table shows the URI parameters for the get flavor disabled status details request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{flavor_id}	Uuid	The ID of the flavor of interest to you.

This operation does not accept a request body.

### 4.4.1.2. Response

#### Example 4.53. Get flavor disabled status details: JSON response

```
{
  "flavor": {
    "OS-FLV-DISABLED:disabled": false,
    "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",
    "ram": 512,
    "vcpus": 1
  }
}
```

#### Example 4.54. Get flavor disabled status details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:OS-FLV-DISABLED="http://docs.openstack.org/compute/ext/
flavor_disabled/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1" disk="1" vcpus="1" ram="512"
name="m1.tiny" id="1" OS-FLV-DISABLED:disabled="False">
  <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
  <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
</flavor>
```

This operation does not return a response body.

## 4.4.2. List flavors with flavor disabled status

Method	URI	Description
GET	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the OS-FLV-DISABLED:disabled attribute.

**Normal response codes:** 200

### 4.4.2.1. Request

This table shows the URI parameters for the list flavors with flavor disabled status 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.

### 4.4.2.2. Response

#### Example 4.55. List flavors with flavor disabled status: JSON response

```
{
  "flavors": [
    {
      "OS-FLV-DISABLED:disabled": false,
      "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",
      "ram": 512,
      "vcpus": 1
    },
    {
      "OS-FLV-DISABLED:disabled": false,
      "disk": 20,
      "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,
    "vcpus": 1
},
{
    "OS-FLV-DISABLED:disabled": false,
    "disk": 40,
    "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,
    "vcpus": 2
},
{
    "OS-FLV-DISABLED:disabled": false,
    "disk": 80,
    "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,
    "vcpus": 4
},
{
    "OS-FLV-DISABLED:disabled": false,
    "disk": 160,
    "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,
    "vcpus": 8
}
]
```

### Example 4.56. List flavors with flavor disabled status: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:OS-FLV-DISABLED="http://docs.openstack.org/compute/ext/
flavor_disabled/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1">
    <flavor disk="1" vcpus="1" ram="512" name="m1.tiny" id="1" OS-FLV-
DISABLED:disabled="False">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <flavor disk="20" vcpus="1" ram="2048" name="m1.small" id="2" OS-FLV-
DISABLED:disabled="False">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/2" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/2" rel=
"bookmark"/>
    </flavor>
    <flavor disk="40" vcpus="2" ram="4096" name="m1.medium" id="3" OS-FLV-
DISABLED:disabled="False">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/3" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/3" rel=
"bookmark"/>
    </flavor>
    <flavor disk="80" vcpus="4" ram="8192" name="m1.large" id="4" OS-FLV-
DISABLED:disabled="False">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/4" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/4" rel=
"bookmark"/>
    </flavor>
    <flavor disk="160" vcpus="8" ram="16384" name="m1.xlarge" id="5" OS-FLV-
DISABLED:disabled="False">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/5" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/5" rel=
"bookmark"/>
    </flavor>
</flavors>
```

This operation does not return a response body.

## 4.5. Flavor extra-specs (os-extra-specs)

List, create, and update the extra-specs or keys for a flavor.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Lists the extra-specs or keys for the specified flavor.
<b>POST</b>	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Creates extra-specs or keys for the specified flavor.
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{key_id}	Gets the value of the specified key.
<b>DELETE</b>	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{key_id}	Deletes a specified extra-spec by key.

## 4.5.1. List flavor extra specs

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Lists the extra-specs or keys for the specified flavor.

Normal response codes: 200

### 4.5.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{flavor_id}	String	The ID of the flavor of interest to you.

This operation does not accept a request body.

### 4.5.1.2. Response

#### Example 4.57. List flavor extra specs: JSON response

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

#### Example 4.58. List flavor extra specs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extra_specs>
    <key2>value2</key2>
    <key1>value1</key1>
</extra_specs>
```

This operation does not return a response body.

## 4.5.2. Create flavor extra specs

Method	URI	Description
POST	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Creates extra-specs or keys for the specified flavor.

**Normal response codes:** 200

### 4.5.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{flavor_id}	String	The ID of the flavor of interest to you.

#### Example 4.59. Create flavor extra specs: JSON request

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

#### Example 4.60. Create flavor extra specs: XML request

```
<?xml version="1.0" encoding="UTF-8" ?>
<extra_specs>
  <key1>value1</key1>
  <key2>value2</key2>
</extra_specs>
```

This operation does not accept a request body.

### 4.5.2.2. Response

#### Example 4.61. Create flavor extra specs: JSON response

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

#### Example 4.62. Create flavor extra specs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extra_specs>
  <key2>value2</key2>
  <key1>value1</key1>
</extra_specs>
```

This operation does not return a response body.

## 4.5.3. Get flavor extra spec details

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{key_id}	Gets the value of the specified key.

Normal response codes: 200

### 4.5.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{flavor_id}	String	The ID of the flavor of interest to you.

This operation does not accept a request body.

### 4.5.3.2. Response

#### Example 4.63. Get flavor extra spec details: JSON response

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

#### Example 4.64. Get flavor extra spec details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<extra_spec key="key1">value1</extra_spec>
```

This operation does not return a response body.

## 4.5.4. Delete flavor extra specs

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{key_id}	Deletes a specified extra-spec by key.

Normal response codes: 200

### 4.5.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{flavor_id}	String	The ID of the flavor of interest to you.

This operation does not accept a request body.

## 4.6. Flavors with rxtx\_factor extended attribute (flavors)

Create, get details for, and list details for flavors. Includes the `rxtx_factor` extended attribute, related to configured bandwidth cap values.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/flavors	Creates a flavor. Includes the <code>rxtx_factor</code> extended attribute.
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the <code>rxtx_factor</code> extended attribute.
<b>GET</b>	/v2/{tenant_id}/flavors/detail	Lists details for available flavors and includes the <code>rxtx_factor</code> extended attribute.

## 4.6.1. Create flavor with rxtx\_factor

Method	URI	Description
POST	/v2/{tenant_id}/flavors	Creates a flavor. Includes the rxtx_factor extended attribute.

Normal response codes: 200

### 4.6.1.1. Request

This table shows the URI parameters for the create flavor with rxtx\_factor request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.65. Create flavor with rxtx\_factor: JSON request

```
{
  "flavor": {
    "name": "flavortest",
    "ram": 1024,
    "vcpus": 2,
    "disk": 10,
    "id": "100",
    "rxtx_factor": 2.0
  }
}
```

#### Example 4.66. Create flavor with rxtx\_factor: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<flavor xmlns="http://docs.openstack.org/compute/api/v1.1"
         xmlns:OS-FLV-EXT-DATA="http://docs.openstack.org/compute/ext/
flavor_extra_data/api/v1.1"
         name="flavortest"
         ram="1024"
         vcpus="2"
         disk="10"
         id="100"
         rxtx_factor="2.0" />
```

This operation does not accept a request body.

### 4.6.1.2. Response

#### Example 4.67. Create flavor with rxtx\_factor: JSON response

```
{
  "flavor": {
    "disk": 10,
    "id": "100",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/flavors/
100",
```

```
        "rel": "self"
    },
    {
        "href": "http://openstack.example.com/openstack/flavors/100",
        "rel": "bookmark"
    }
],
"name": "flavortest",
"ram": 1024,
"rxtx_factor": 2.0,
"vcpus": 2
}
}
```

### Example 4.68. Create flavor with rxtx\_factor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" disk="10" vcpus="2" ram="1024" name="flavortest" id="100" rxtx_factor="2.0">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/100" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/100" rel="bookmark"/>
</flavor>
```

This operation does not return a response body.

## 4.6.2. Get flavor with rtx\_factor

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the rtx_factor extended attribute.

Normal response codes: 200200

### 4.6.2.1. Request

This table shows the URI parameters for the get flavor with rtx\_factor request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{flavor_id}	String	The flavor ID.

This operation does not accept a request body.

### 4.6.2.2. Response

#### Example 4.69. Get flavor with rtx\_factor: JSON response

```
{
  "flavor": {
    "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",
    "ram": 512,
    "rtx_factor": 1.0,
    "vcpus": 1
  }
}
```

#### Example 4.70. Get flavor with rtx\_factor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" disk="1" vcpus="1" ram="512" name="m1.tiny" id="1" rtx_factor="1.0">
  <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/flavors/1" rel="bookmark"/>
</flavor>
```

This operation does not return a response body.

## 4.6.3. Get flavor Details with rxtx\_factor

Method	URI	Description
GET	/v2/{tenant_id}/flavors/detail	Lists details for available flavors and includes the rxtx_factor extended attribute.

**Normal response codes:** 200

### 4.6.3.1. Request

This table shows the URI parameters for the get flavor details with rxtx\_factor 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.

### 4.6.3.2. Response

#### Example 4.71. Get flavor Details with rxtx\_factor: JSON response

```
{
    "flavors": [
        {
            "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",
            "ram": 512,
            "rxtx_factor": 1.0,
            "vcpus": 1
        },
        {
            "disk": 20,
            "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,
        "vcpus": 1
    },
    {
        "disk": 40,
        "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,
        "vcpus": 2
    },
    {
        "disk": 80,
        "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,
        "vcpus": 4
    },
    {
        "disk": 160,
        "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,
        "vcpus": 8
    }
}
```

### Example 4.72. Get flavor Details with rxtx\_factor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
    <flavor disk="1" vcpus="1" ram="512" name="m1.tiny" id="1" rxtx_factor="1.
0">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <flavor disk="20" vcpus="1" ram="2048" name="m1.small" id="2" rxtx_factor=
"1.0">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/2" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/2" rel=
"bookmark"/>
    </flavor>
    <flavor disk="40" vcpus="2" ram="4096" name="m1.medium" id="3" rxtx_factor=
"1.0">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/3" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/3" rel=
"bookmark"/>
    </flavor>
    <flavor disk="80" vcpus="4" ram="8192" name="m1.large" id="4" rxtx_factor=
"1.0">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/4" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/4" rel=
"bookmark"/>
    </flavor>
    <flavor disk="160" vcpus="8" ram="16384" name="m1.xlarge" id="5"
rxtx_factor="1.0">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/5" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/5" rel=
"bookmark"/>
    </flavor>
</flavors>
```

This operation does not return a response body.

## 4.7. Flavors with extended attributes (flavors)

Create, get details for, and list details for flavors. Includes the `rxtx_factor`, `OS-FLV-EXT-DATA:ephemeral`, and `swap` extended attributes.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/flavors	Creates a flavor. Includes the rtx_factor, OS-FLV-EXT- DATA:ephemeral, and swap extended attributes.
<b>GET</b>	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the rtx_factor, OS-FLV-EXT- DATA:ephemeral, and swap extended attributes.
<b>GET</b>	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the rtx_factor, OS-FLV-EXT- DATA:ephemeral, and swap extended attributes.

## 4.7.1. Create flavor with extra data

Method	URI	Description
POST	/v2/{tenant_id}/flavors	Creates a flavor. Includes the rxtx_factor, OS-FLV-EXT- DATA:ephemeral, and swap extended attributes.

**Normal response codes:** 200

### 4.7.1.1. Request

This table shows the URI parameters for the create flavor with extra data request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.73. Create flavor with extra data: JSON request

```
{
  "flavor": {
    "name": "flavortest",
    "ram": 1024,
    "vcpus": 2,
    "disk": 10,
    "id": "666",
    "rxtx_factor": 2.0,
    "OS-FLV-EXT-DATA:ephemeral": 30,
    "swap": 5
  }
}
```

This table shows the body parameters for the create flavor with extra data request:

Name	Type	Description
name	(Required)	The name for the flavor.
ram	(Required)	The amount of RAM, in MBs, for the flavor.
vcpus	(Required)	The virtual CPUs, in a whole integer amount, for the flavor.
disk	(Required)	The amount of disk space, in GBs, for the flavor.
id	(Required)	The ID, a unique integer value, for the flavor.
rxtx_factor	(Required)	Enables you to override the bandwidth cap value, which defaults to the bandwidth cap value for the network to which the flavor is attached. The new bandwidth cap is calculated by multiplying the rxtx_factor value that you specify by the rxtx_base property for the network.
OS-FLV-EXT-DATA:ephemeral	(Required)	The size of secondary drive, in GBs, to attach to a server.
swap	(Required)	The amount of swap disk space, in GBs, to allocate to a server.

#### Example 4.74. Create flavor with extra data: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<flavor xmlns="http://docs.openstack.org/compute/api/v1.1"
         xmlns:OS-FLV-EXT-DATA="http://docs.openstack.org/compute/ext/
flavor_extra_data/api/v1.1"
         name="flavortest"
```

```
    ram="1024"
    vcpus="2"
    disk="10"
    id="666"
    swap="5"
    rxtx_factor="2.0"
    OS-FLV-EXT-DATA:ephemeral="30" />
```

This operation does not accept a request body.

#### 4.7.1.2. Response

##### Example 4.75. Create flavor with extra data: JSON response

```
{
  "flavor": {
    "OS-FLV-EXT-DATA:ephemeral": 30,
    "disk": 10,
    "id": "666",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/flavors/666",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/666",
        "rel": "bookmark"
      }
    ],
    "name": "flavortest",
    "ram": 1024,
    "vcpus": 2
  }
}
```

##### Example 4.76. Create flavor with extra data: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:OS-FLV-EXT-DATA="http://docs.openstack.org/compute/ext/
flavor_extra_data/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1" disk="10" vcpus="2" ram="1024"
name="flavortest" id="666" OS-FLV-EXT-DATA:ephemeral="30">
  <atom:link href="http://openstack.example.com/v2/openstack/flavors/666" rel=
"self"/>
  <atom:link href="http://openstack.example.com/openstack/flavors/666" rel=
"bookmark"/>
</flavor>
```

This operation does not return a response body.

## 4.7.2. Get flavor extra data details

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.

Normal response codes: 200

### 4.7.2.1. Request

This table shows the URI parameters for the get flavor extra data details request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{flavor_id}	String	The ID of the flavor of interest to you.

This operation does not accept a request body.

### 4.7.2.2. Response

#### Example 4.77. Get flavor extra data details: JSON response

```
{
  "flavor": {
    "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",
    "ram": 512,
    "vcpus": 1
  }
}
```

#### Example 4.78. Get flavor extra data details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:OS-FLV-EXT-DATA="http://docs.openstack.org/compute/ext/
flavor_extra_data/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1" disk="1" vcpus="1" ram="512"
name="m1.tiny" id="1" OS-FLV-EXT-DATA:ephemeral="0">
  <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
  <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
```

```
</flavor>
```

This operation does not return a response body.

## 4.7.3. List flavors with extra data

Method	URI	Description
GET	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the rxtx_factor, OS-FLV-EXT- DATA:ephemeral, and swap extended attributes.

**Normal response codes:** 200

### 4.7.3.1. Request

This table shows the URI parameters for the list flavors with extra data 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.

### 4.7.3.2. Response

#### Example 4.79. List flavors with extra data: JSON response

```
{
    "flavors": [
        {
            "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",
            "ram": 512,
            "vcpus": 1
        },
        {
            "OS-FLV-EXT-DATA:ephemeral": 0,
            "disk": 20,
            "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,
    "vcpus": 1
},
{
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "disk": 40,
    "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,
    "vcpus": 2
},
{
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "disk": 80,
    "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,
    "vcpus": 4
},
{
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "disk": 160,
    "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,
    "vcpus": 8
}
}
```

### Example 4.80. List flavors with extra data: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:OS-FLV-EXT-DATA="http://docs.openstack.org/compute/ext/
flavor_extra_data/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1">
    <flavor disk="1" vcpus="1" ram="512" name="m1.tiny" id="1" OS-FLV-EXT-
DATA:ephemeral="0">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <flavor disk="20" vcpus="1" ram="2048" name="m1.small" id="2" OS-FLV-EXT-
DATA:ephemeral="0">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/2" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/2" rel=
"bookmark"/>
    </flavor>
    <flavor disk="40" vcpus="2" ram="4096" name="m1.medium" id="3" OS-FLV-EXT-
DATA:ephemeral="0">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/3" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/3" rel=
"bookmark"/>
    </flavor>
    <flavor disk="80" vcpus="4" ram="8192" name="m1.large" id="4" OS-FLV-EXT-
DATA:ephemeral="0">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/4" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/4" rel=
"bookmark"/>
    </flavor>
    <flavor disk="160" vcpus="8" ram="16384" name="m1.xlarge" id="5" OS-FLV-EXT-
DATA:ephemeral="0">
        <atom:link href="http://openstack.example.com/v2/openstack/flavors/5" rel=
"self"/>
        <atom:link href="http://openstack.example.com/openstack/flavors/5" rel=
"bookmark"/>
    </flavor>
</flavors>
```

This operation does not return a response body.

## 4.8. Flavors create or delete (flavors)

Create or delete flavors.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/flavors	Creates a flavor.
<b>DELETE</b>	/v2/{tenant_id}/flavors/{flavor_id}	Deletes a flavor.

## 4.8.1. Create flavor

Method	URI	Description
POST	/v2/{tenant_id}/flavors	Creates a flavor.

**Normal response codes:** 200

### 4.8.1.1. Request

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

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

#### Example 4.81. Create flavor: JSON request

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

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

Name	Type	Description
name	(Required)	The name for the new flavor.
ram	(Required)	The amount of RAM, in MBs, for this flavor.
vcpus	(Required)	The virtual CPUs, in a whole integer amount, for the new flavor.
disk	(Required)	The amount of disk space, in GBs.
id	(Required)	The ID for the new flavor, which is a unique integer value.

#### Example 4.82. Create flavor: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<flavor>
  <name>test_flavor</name>
  <ram>1024</ram>
  <vcpus>2</vcpus>
  <disk>10</disk>
  <id>10</id>
</flavor>
```

This operation does not accept a request body.

### 4.8.1.2. Response

#### Example 4.83. Create flavor: JSON response

```
{
```

```
"flavor": {
    "disk": 10,
    "id": "10",
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/flavors/10",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/flavors/10",
            "rel": "bookmark"
        }
    ],
    "name": "test_flavor",
    "ram": 1024,
    "vcpus": 2
}
```

#### Example 4.84. Create flavor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" disk="10" vcpus="2" ram="1024" name="test_flavor" id="10">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/10" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/10" rel="bookmark"/>
</flavor>
```

This operation does not return a response body.

## 4.8.2. Delete flavor

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/flavors/{flavor_id}	Deletes a flavor.

**Normal response codes:** 204

### 4.8.2.1. Request

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

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

This operation does not accept a request body.

## 4.9. Images with size attribute (images)

List details for available images or get details for a specified image. Includes the OS-EXT-IMG-SIZE:size extended attribute, which shows the image size.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/images/detail	Lists details for available images. Includes the image size.
<b>GET</b>	/v2/{tenant_id}/images/{image_id}	Gets details for a specified image. Includes the image size.

## 4.9.1. List details for images

Method	URI	Description
GET	/v2/{tenant_id}/images/detail	Lists details for available images. Includes the image size.

**Normal response codes:** 200200

### 4.9.1.1. Request

This table shows the URI parameters for the list details for images 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.

### 4.9.1.2. Response

#### Example 4.85. List details for images: 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/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        },
        {
          "href": "http://glance.openstack.example.com/openstack/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": "25165824",  
    "created": "2011-01-01T01:02:03Z",  
    "id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",  
    "links": [  
        {  
            "href": "http://openstack.example.com/v2/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",  
            "rel": "self"  
        },  
        {  
            "href": "http://openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",  
            "rel": "bookmark"  
        },  
        {  
            "href": "http://glance.openstack.example.com/openstack/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"  
},  
{  
    "OS-EXT-IMG-SIZE:size": "58145823",  
    "created": "2011-01-01T01:02:03Z",  
    "id": "a2459075-d96c-40d5-893e-577ff92e721c",  
    "links": [  
        {  
            "href": "http://openstack.example.com/v2/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",  
            "rel": "self"  
        },  
        {  
            "href": "http://openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",  
            "rel": "bookmark"  
        },  
        {  
            "href": "http://glance.openstack.example.com/openstack/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"
    },
    {
        "OS-EXT-IMG-SIZE:size": "49163826",
        "created": "2011-01-01T01:02:03Z",
        "id": "a440c04b-79fa-479c-bed1-0b816eaec379",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/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"
    },
    {
        "OS-EXT-IMG-SIZE:size": "26360814",
        "created": "2011-01-01T01:02:03Z",
        "id": "c905cedb-7281-47e4-8a62-f26bc5fc4c77",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "alternate"
            }
        ]
    }
]
```

```
        "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"
},
{
    "OS-EXT-IMG-SIZE:size": "84035174",
    "created": "2011-01-01T01:02:03Z",
    "id": "cedef40a-ed67-4d10-800e-17455edce175",
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/images/cedef40a-ed67-4d10-800e-17455edce175",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/images/cedef40a-ed67-4d10-800e-17455edce175",
            "rel": "bookmark"
        },
        {
            "href": "http://glance.openstack.example.com/openstack/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"
},
{
    "OS-EXT-IMG-SIZE:size": "83594576",
    "created": "2011-01-01T01:02:03Z",
    "id": "76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
            "rel": "self"
        },
        {

```

```

        "href": "http://openstack.example.com/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
        "rel": "bookmark"
    },
    {
        "href": "http://glance.openstack.example.com/openstack/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 4.86. List details for images: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<images xmlns:OS-EXT-IMG-SIZE="http://docs.openstack.org/compute/ext/
image_size/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/compute/api/v1.1">
    <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage7"
    created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
    "70a599e0-31e7-49b7-b260-868f441e862b" OS-EXT-IMG-SIZE:size="74185822">
        <metadata>
            <meta key="kernel_id">nokernel</meta>
            <meta key="auto_disk_config">True</meta>
            <meta key="ramdisk_id">nokernel</meta>
            <meta key="architecture">x86_64</meta>
        </metadata>
        <atom:link href="http://openstack.example.com/v2/openstack/images/
    70a599e0-31e7-49b7-b260-868f441e862b" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/images/
    70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        <atom:link href="http://glance.openstack.example.com/openstack/images/
    70a599e0-31e7-49b7-b260-868f441e862b" type="application/vnd.openstack.image"
    rel="alternate"/>
    </image>
    <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
    created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
    "155d900f-4e14-4e4c-a73d-069cbf4541e6" OS-EXT-IMG-SIZE:size="25165824">
        <metadata>
            <meta key="kernel_id">nokernel</meta>
            <meta key="ramdisk_id">nokernel</meta>
            <meta key="architecture">x86_64</meta>
        </metadata>
        <atom:link href="http://openstack.example.com/v2/openstack/images/
    155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/images/
    155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="bookmark"/>

```

```
<atom:link href="http://glance.openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="a2459075-d96c-40d5-893e-577ff92e721c" OS-EXT-IMG-SIZE:size="58145823">
<metadata>
<meta key="kernel_id">nokernel</meta>
<meta key="ramdisk_id">nokernel</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage6" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="a440c04b-79fa-479c-bed1-0b816eaec379" OS-EXT-IMG-SIZE:size="49163826">
<metadata>
<meta key="kernel_id">nokernel</meta>
<meta key="auto_disk_config">False</meta>
<meta key="ramdisk_id">nokernel</meta>
<meta key="architecture">x86_64</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="c905cedb-7281-47e4-8a62-f26bc5fc4c77" OS-EXT-IMG-SIZE:size="26360814">
<metadata>
<meta key="kernel_id">155d900f-4e14-4e4c-a73d-069cbf4541e6</meta>
<meta key="ramdisk_id">None</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="cedef40a-ed67-4d10-800e-17455edce175" OS-EXT-IMG-SIZE:size="84035174">
<metadata>
<meta key="kernel_id">nokernel</meta>
<meta key="ramdisk_id">nokernel</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/cedef40a-ed67-4d10-800e-17455edce175" rel="self"/>
```

```
<atom:link href="http://openstack.example.com/openstack/images/cedef40a-
ed67-4d10-800e-17455edce175" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175" type="application/vnd.openstack.image"
rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" OS-EXT-IMG-SIZE:size="83594576">
<metadata>
<meta key="kernel_id">nokernel</meta>
<meta key="ramdisk_id">nokernel</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/76fa36fc-
c930-4bf3-8c8a-ea2a2420deb6" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" type="application/vnd.openstack.image"
rel="alternate"/>
</image>
</images>
```

This operation does not return a response body.

## 4.9.2. Get image details

Method	URI	Description
GET	/v2/{tenant_id}/images/{image_id}	Gets details for a specified image. Includes the image size.

**Normal response codes:** 200200

### 4.9.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{image_id}	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

### 4.9.2.2. Response

#### Example 4.87. 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/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "bookmark"
      },
      {
        "href": "http://glance.openstack.example.com/openstack/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 4.88. Get image details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<image xmlns:OS-EXT-IMG-SIZE="http://docs.openstack.org/compute/ext/
image_size/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1" status="ACTIVE" updated=
"2011-01-01T01:02:03Z" name="fakeimage7" created="2011-01-01T01:02:03Z"
minDisk="0" progress="100" minRam="0" id="70a599e0-31e7-49b7-
b260-868f441e862b" OS-EXT-IMG-SIZE:size="74185822">
    <metadata>
        <meta key="kernel_id">nokernel</meta>
        <meta key="auto_disk_config">True</meta>
        <meta key="ramdisk_id">nokernel</meta>
        <meta key="architecture">x86_64</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" type="application/vnd.openstack.image"
rel="alternate"/>
</image>
```

This operation does not return a response body.

## 4.10. Limits with project usage (limits)

Extend limits to show the project usage. Show information such as RAM or instance quotas usage.

Method	URI	Description
GET	/v2/{tenant_id}/limits	Gets absolute and rate limit information, including information on currently used absolute limits.

## 4.10.1. Get limits

Method	URI	Description
GET	/v2/{tenant_id}/limits	Gets absolute and rate limit information, including information on currently used absolute limits.

Normal response codes: 200

### 4.10.1.1. Request

This table shows the URI parameters for the get limits 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.

### 4.10.1.2. Response

#### Example 4.89. Get 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,
      "totalCoresUsed": 0,
      "totalInstancesUsed": 0,
      "totalRAMUsed": 0,
      "totalSecurityGroupsUsed": 0,
      "totalFloatingIpsUsed": 0
    },
    "rate": [
      {
        "limit": [
          {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "POST"
          },
          {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "PUT"
          }
        ]
      }
    ]
  }
}
```

```
        "value": 120,
        "verb": "PUT"
    },
    {
        "next-available": "2012-11-27T17:24:52Z",
        "remaining": 120,
        "unit": "MINUTE",
        "value": 120,
        "verb": "DELETE"
    }
],
"regex": ".*",
"uri": "*"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "POST"
        }
],
"regex": "^/servers",
"uri": "*/servers"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "GET"
        }
],
"regex": ".*changes-since.*",
"uri": "*changes-since"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 12,
            "unit": "MINUTE",
            "value": 12,
            "verb": "GET"
        }
],
"regex": "^/os-fping",
"uri": "*/os-fping"
}
]
```

#### Example 4.90. Get limits: XML response

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

```

<limits xmlns:os-used-limits="http://docs.openstack.org/compute/ext/
used_limits/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/common/api/v1.0">
    <rates>
        <rate regex=".*" uri="*">
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="POST"
remaining="120" value="120"/>
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="PUT"
remaining="120" value="120"/>
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="DELETE"
remaining="120" value="120"/>
        </rate>
        <rate regex="^/servers" uri="*/servers">
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="POST"
remaining="120" value="120"/>
        </rate>
        <rate regex=".*changes-since.*" uri="*changes-since*>">
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="GET"
remaining="120" value="120"/>
        </rate>
        <rate regex="^/os-fping" uri="*/os-fping">
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="GET"
remaining="12" value="12"/>
        </rate>
    </rates>
    <absolute>
        <limit name="maxServerMeta" value="128"/>
        <limit name="maxPersonality" value="5"/>
        <limit name="maxImageMeta" value="128"/>
        <limit name="maxPersonalitySize" value="10240"/>
        <limit name="maxSecurityGroupRules" value="20"/>
        <limit name="maxTotalKeypairs" value="100"/>
        <limit name="totalRAMUsed" value="0"/>
        <limit name="totalInstancesUsed" value="0"/>
        <limit name="maxSecurityGroups" value="10"/>
        <limit name="totalFloatingIpsUsed" value="0"/>
        <limit name="maxTotalCores" value="20"/>
        <limit name="totalSecurityGroupsUsed" value="0"/>
        <limit name="maxTotalFloatingIps" value="10"/>
        <limit name="maxTotalInstances" value="10"/>
        <limit name="totalCoresUsed" value="0"/>
        <limit name="maxTotalRAMSize" value="51200"/>
    </absolute>
</limits>

```

This operation does not return a response body.

## 4.11. Limits with project usage for administrators (limits)

Extend limits to enable administrators to show the project usage for a specified customer project ID. Show information such as RAM or instance quotas usage.

Method	URI	Description
GET	/v2/{tenant_id}/limits/{tenant_id}	Enables administrators to get absolute and rate limit information, including information about currently used absolute limits, for a specified customer tenant ID.



## 4.11.1. Get customer limits

Method	URI	Description
GET	/v2/{tenant_id}/limits/{tenant_id}	Enables administrators to get absolute and rate limit information, including information about currently used absolute limits, for a specified customer tenant ID.

**Normal response codes:** 200

### 4.11.1.1. Request

This table shows the URI parameters for the get customer limits request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{tenant_id}	Uuid	The ID for the tenant for which an administrator wants to get quota information. Specify the customer tenant ID in the URI as tenant_id={tenant_id}.

This operation does not accept a request body.

### 4.11.1.2. Response

#### Example 4.91. Used limits for admins: 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,
            "totalCoresUsed": 0,
            "totalInstancesUsed": 0,
            "totalRAMUsed": 0,
            "totalSecurityGroupsUsed": 0,
            "totalFloatingIpsUsed": 0
        },
        "rate": [
            {
                "limit": [
                    {
                        "next-available": "2012-11-27T17:24:52Z",
                        "remaining": 120,
                        "unit": "MINUTE",
                        "value": 120,
                        "verb": "POST"
                    },
                    {
                        "next-available": "2012-11-27T17:24:52Z",
                        "remaining": 120,
                        "unit": "MINUTE",
                        "value": 120,
                        "verb": "PUT"
                    }
                ]
            }
        ]
    }
}
```

```
        "remaining": 120,
        "unit": "MINUTE",
        "value": 120,
        "verb": "PUT"
    },
    {
        "next-available": "2012-11-27T17:24:52Z",
        "remaining": 120,
        "unit": "MINUTE",
        "value": 120,
        "verb": "DELETE"
    }
],
"regex": ".*",
"uri": "*"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "POST"
        }
    ],
    "regex": "^/servers",
    "uri": "*/servers"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "GET"
        }
    ],
    "regex": ".*changes-since.*",
    "uri": "*/changes-since"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 12,
            "unit": "MINUTE",
            "value": 12,
            "verb": "GET"
        }
    ],
    "regex": "^/os-fping",
    "uri": "*/os-fping"
}
]
}
```

### Example 4.92. Used limits for admins: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<limits xmlns:os-used-limits="http://docs.openstack.org/compute/ext/
used_limits/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/common/api/v1.0">
    <rates>
        <rate regex=".*" uri="*.*">
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="POST"
            remaining="120" value="120"/>
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="PUT"
            remaining="120" value="120"/>
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="DELETE"
            remaining="120" value="120"/>
        </rate>
        <rate regex="^/servers" uri="*/servers">
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="POST"
            remaining="120" value="120"/>
        </rate>
        <rate regex=".changes-since.*" uri="*changes-since*>">
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="GET"
            remaining="120" value="120"/>
        </rate>
        <rate regex="^/os-fping" uri="*/os-fping">
            <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="GET"
            remaining="12" value="12"/>
        </rate>
    </rates>
    <absolute>
        <limit name="maxServerMeta" value="128"/>
        <limit name="maxPersonality" value="5"/>
        <limit name="maxImageMeta" value="128"/>
        <limit name="maxPersonalitySize" value="10240"/>
        <limit name="maxSecurityGroupRules" value="20"/>
        <limit name="maxTotalKeypairs" value="100"/>
        <limit name="totalRAMUsed" value="0"/>
        <limit name="totalInstancesUsed" value="0"/>
        <limit name="maxSecurityGroups" value="10"/>
        <limit name="totalFloatingIpsUsed" value="0"/>
        <limit name="maxTotalCores" value="20"/>
        <limit name="totalSecurityGroupsUsed" value="0"/>
        <limit name="maxTotalFloatingIps" value="10"/>
        <limit name="maxTotalInstances" value="10"/>
        <limit name="totalCoresUsed" value="0"/>
        <limit name="maxTotalRAMSize" value="51200"/>
    </absolute>
</limits>
```

This operation does not return a response body.

## 4.12. Guest agents (os-agents)

Create, update, and delete guest agents. Use guest agents to access files on the disk, configure networking, or run other applications or scripts in the guest while it 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
<b>GET</b>	/v2/{tenant_id}/os-agents	Lists all agent builds.
<b>POST</b>	/v2/{tenant_id}/os-agents	Creates an agent build.
<b>DELETE</b>	/v2/{tenant_id}/os-agents	Deletes an existing agent build.
<b>PUT</b>	/v2/{tenant_id}/os-agents/{id}	Updates an agent build.

## 4.12.1. List agents

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

**Normal response codes:** 200

### 4.12.1.1. Request

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

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

This operation does not accept a request body.

### 4.12.1.2. Response

#### Example 4.93. List agents: JSON response

```
{
  "agents": [
    {
      "agent_id": "1",
      "architecture": "x86",
      "hypervisor": "hypervisor",
      "md5hash": "add6bb58e139be103324d04d82d8f545",
      "os": "os",
      "url": "xxxxxxxxxxxxxx",
      "version": "8.0"
    }
  ]
}
```

#### Example 4.94. List agents: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<agents>
  <agent url="xxxxxxxxxxxxxx" hypervisor="hypervisor" md5hash=
"add6bb58e139be103324d04d82d8f545" version="8.0" architecture="x86" os="os"
  agent_id="1"/>
</agents>
```

This operation does not return a response body.

## 4.12.2. Create agent

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

**Normal response codes:** 200

### 4.12.2.1. Request

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

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

#### Example 4.95. Create agent: JSON request

```
{
  "agent": {
    "hypervisor": "hypervisor",
    "os": "os",
    "architecture": "x86",
    "version": "8.0",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "url": "xxxxxxxxxxxx"
  }
}
```

#### Example 4.96. Create agent: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<agent>
  <hypervisor>hypervisor</hypervisor>
  <os>os</os>
  <architecture>x86</architecture>
  <version>8.0</version>
  <md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
  <url>xxxxxxxxxxxx</url>
</agent>
```

This operation does not accept a request body.

### 4.12.2.2. Response

#### Example 4.97. Create agent: JSON response

```
{
  "agent": {
    "agent_id": "1",
    "architecture": "x86",
    "hypervisor": "hypervisor",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "os": "os",
    "url": "xxxxxxxxxxxx",
    "version": "8.0"
  }
}
```

{

**Example 4.98. Create agent: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<agent>
  <url>xxxxxxxxxxxx</url>
  <hypervisor>hypervisor</hypervisor>
  <md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
  <version>8.0</version>
  <architecture>x86</architecture>
  <os>os</os>
  <agent_id>1</agent_id>
</agent>
```

This operation does not return a response body.

## 4.12.3. Delete agent

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

**Normal response codes:** 202

### 4.12.3.1. Request

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

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

This operation does not accept a request body.

## 4.12.4. Update agent

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

**Normal response codes:** 200

### 4.12.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique ID of the tenant or account.
{id}	UUID	The unique ID associated with the agent.

#### Example 4.99. Update agent: JSON request

```
{
  "para": {
    "url": "xxx://xxxx/xxx/xxx",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "version": "7.0"
  }
}
```

#### Example 4.100. Update agent: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<para>
  <version>7.0</version>
  <url>xxx://xxxx/xxx/xxx</url>
  <md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
</para>
```

This operation does not accept a request body.

### 4.12.4.2. Response

#### Example 4.101. Update agent: JSON response

```
{
  "agent": {
    "agent_id": "1",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "url": "xxx://xxxx/xxx/xxx",
    "version": "7.0"
  }
}
```

#### Example 4.102. Update agent: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<agent>
  <url>xxx://xxxx/xxx/xxx</url>
```

```
<version>7.0</version>
<agent_id>1</agent_id>
<md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
</agent>
```

This operation does not return a response body.

## 4.13. Host aggregates (os-aggregates)

Create and manage host aggregates. An aggregate assigns metadata to groups of compute nodes. Aggregates are only visible to the cloud provider.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-aggregates	Lists all aggregates.
<b>POST</b>	/v2/{tenant_id}/os-aggregates	Creates an aggregate.
<b>DELETE</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Deletes an aggregate.
<b>GET</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Gets details about a specified aggregate.
<b>PUT</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Updates the name, and optionally the availability zone, for a specified aggregate.
<b>POST</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Sets metadata for an aggregate.
<b>POST</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Adds a host to an aggregate.
<b>POST</b>	/v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Removes a host from an aggregate.

## 4.13.1. List aggregates

Method	URI	Description
GET	/v2/{tenant_id}/os-aggregates	Lists all aggregates.

**Normal response codes:** 200

### 4.13.1.1. Request

This table shows the URI parameters for the list aggregates 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.

### 4.13.1.2. Response

#### Example 4.103. List aggregates: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregates>
  <aggregate>
    <name>name</name>
    <availability_zone>nova</availability_zone>
    <deleted>False</deleted>
    <created_at>2012-11-16 06:22:25.935099</created_at>
    <updated_at>None</updated_at>
    <hosts/>
    <deleted_at>None</deleted_at>
    <id>1</id>
    <metadata>
      <availability_zone>nova</availability_zone>
    </metadata>
  </aggregate>
</aggregates>
```

#### Example 4.104. List aggregates: JSON response

```
{
  "aggregates": [
    {
      "availability_zone": "nova",
      "created_at": "2012-11-16T06:22:23.361359",
      "deleted": false,
      "deleted_at": null,
      "hosts": [],
      "id": 1,
      "metadata": {
        "availability_zone": "nova"
      },
      "name": "name",
      "updated_at": null
    }
  ]
}
```

{}

This operation does not return a response body.

## 4.13.2. Create aggregate

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

**Normal response codes:** 200200

### 4.13.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.105. Create aggregate: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<aggregate name="name" availability_zone="nova" />
```

#### Example 4.106. Create aggregate: JSON request

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

This operation does not accept a request body.

### 4.13.2.2. Response

#### Example 4.107. Create aggregate: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
    <name>name</name>
    <availability_zone>nova</availability_zone>
    <deleted>False</deleted>
    <created_at>2012-10-01 18:50:35.506667</created_at>
    <updated_at>None</updated_at>
    <deleted_at>None</deleted_at>
    <id>1</id>
</aggregate>
```

#### Example 4.108. Create aggregate: JSON response

```
{
    "aggregate": {
        "availability_zone": "nova",
        "created_at": "2012-10-01T18:50:27.781065",
        "deleted": false,
        "deleted_at": null,
        "id": 1,
```

```
        "name": "name",
        "updated_at": null
    }
```

This operation does not return a response body.

## 4.13.3. Delete aggregate

Method	URI	Description
DELETE	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Deletes an aggregate.

**Normal response codes:** 200

### 4.13.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

This operation does not accept a request body.

## 4.13.4. Get aggregate details

Method	URI	Description
GET	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Gets details about a specified aggregate.

Normal response codes: 200200

### 4.13.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

This operation does not accept a request body.

### 4.13.4.2. Response

#### Example 4.109. Get aggregate details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
  <name>name</name>
  <availability_zone>nova</availability_zone>
  <deleted>False</deleted>
  <created_at>2012-11-16 06:22:25.587739</created_at>
  <updated_at>None</updated_at>
  <hosts/>
  <deleted_at>None</deleted_at>
  <id>1</id>
  <metadata>
    <availability_zone>nova</availability_zone>
  </metadata>
</aggregate>
```

#### Example 4.110. Get aggregate details: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2012-11-16T06:22:23.032493",
    "deleted": false,
    "deleted_at": null,
    "hosts": [],
    "id": 1,
    "metadata": {
      "availability_zone": "nova"
    },
    "name": "name",
    "updated_at": null
  }
}
```

This operation does not return a response body.

## 4.13.5. Update aggregate

Method	URI	Description
PUT	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Updates the name, and optionally the availability zone, for a specified aggregate.

**Normal response codes:** 200

### 4.13.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

#### Example 4.111. Update aggregate: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<aggregate name="newname" availability_zone="nova2" />
```

#### Example 4.112. Update aggregate: JSON request

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

This operation does not accept a request body.

### 4.13.5.2. Response

#### Example 4.113. Update aggregate: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
    <name>newname</name>
    <availability_zone>nova2</availability_zone>
    <deleted>False</deleted>
    <created_at>2012-12-04 12:04:30.245284</created_at>
    <updated_at>2012-12-04 12:04:30.357795</updated_at>
    <hosts/>
    <deleted_at>None</deleted_at>
    <id>1</id>
    <metadata>
        <availability_zone>nova2</availability_zone>
    </metadata>
</aggregate>
```

#### Example 4.114. Update aggregate: JSON response

```
{
```

```
    "aggregate": {
        "availability_zone": "nova2",
        "created_at": "2012-12-04T12:04:27.075065",
        "deleted": false,
        "deleted_at": null,
        "hosts": [],
        "id": 1,
        "metadata": {
            "availability_zone": "nova2"
        },
        "name": "newname",
        "updated_at": "2012-12-04T12:04:27.242597"
    }
}
```

This operation does not return a response body.

## 4.13.6. Set aggregate metadata

Method	URI	Description
POST	/v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Sets metadata for an aggregate.

Normal response codes: 200

### 4.13.6.1. Request

This table shows the URI parameters for the set aggregate metadata request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

This table shows the body parameters for the set aggregate metadata request:

Name	Type	Description
metadata	String <i>(Required)</i>	Metadata value to be set.

#### Example 4.115. Set aggregate metadata: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<set_metadata>
  <metadata>
    <key>value</key>
  </metadata>
</set_metadata>
```

#### Example 4.116. Set aggregate metadata: JSON request

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

This operation does not accept a request body.

### 4.13.6.2. Response

#### Example 4.117. Set aggregate metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
<name>name</name>
<availability_zone>nova</availability_zone>
<deleted>False</deleted>
<created_at>2012-11-16 06:22:24.864471</created_at>
```

```
<updated_at>None</updated_at>
<hosts/>
<deleted_at>None</deleted_at>
<id>1</id>
<metadata>
  <key>value</key>
  <availability_zone>nova</availability_zone>
</metadata>
</aggregate>
```

### Example 4.118. Set aggregate metadata: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2012-11-16T06:22:22.342791",
    "deleted": false,
    "deleted_at": null,
    "hosts": [],
    "id": 1,
    "metadata": {
      "availability_zone": "nova",
      "key": "value"
    },
    "name": "name",
    "updated_at": null
  }
}
```

This operation does not return a response body.

## 4.13.7. Add host to aggregate

Method	URI	Description
POST	/v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Adds a host to an aggregate.

Normal response codes: 200

### 4.13.7.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

This table shows the body parameters for the add host to aggregate request:

Name	Type	Description
host	UUID <i>(Required)</i>	Host ID to add to an aggregate, which is a collection of multiple groups of hosts that share common resources like storage and network.

#### Example 4.119. Add host to aggregate: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<add_host host="7c9e00dbca5e4fb88538b021c0f933a5" />
```

#### Example 4.120. Add host to aggregate: JSON request

```
{
    "add_host": {
        "host": "581d29b9e3504d8a895caddb13839b15"
    }
}
```

This operation does not accept a request body.

### 4.13.7.2. Response

#### Example 4.121. Add host to aggregate: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
    <name>name</name>
    <availability_zone>nova</availability_zone>
    <deleted>False</deleted>
    <created_at>2012-12-04 12:04:27.574038</created_at>
    <updated_at>None</updated_at>
    <hosts>
        <host>392adba19dd449179804eaff16ff4a97</host>
    </hosts>
    <deleted_at>None</deleted_at>
    <id>1</id>
```

```
<metadata>
  <availability_zone>nova</availability_zone>
</metadata>
</aggregate>
```

### Example 4.122. Add host to aggregate: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2012-12-04T12:04:24.399784",
    "deleted": false,
    "deleted_at": null,
    "hosts": [
      "0438c6a4e8d841ad823b801d681f4680"
    ],
    "id": 1,
    "metadata": {
      "availability_zone": "nova"
    },
    "name": "name",
    "updated_at": null
  }
}
```

This operation does not return a response body.

## 4.13.8. Remove host from aggregate

Method	URI	Description
POST	/v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Removes a host from an aggregate.

Normal response codes: 200

### 4.13.8.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

This table shows the body parameters for the remove host from aggregate request:

Name	Type	Description
host	UUID <i>(Required)</i>	Host ID to add to an aggregate.

#### Example 4.123. Remove host from aggregate: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<remove_host host="7c9e00dbc5e4fb88538b021c0f933a5" />
```

#### Example 4.124. Remove host from aggregate: JSON request

```
{
    "remove_host": {
        "host": "581d29b9e3504d8a895caddb13839b15"
    }
}
```

This operation does not accept a request body.

### 4.13.8.2. Response

#### Example 4.125. Remove host from aggregate: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
    <name>name</name>
    <availability_zone>nova</availability_zone>
    <deleted>False</deleted>
    <created_at>2012-12-04 12:04:29.722109</created_at>
    <updated_at>None</updated_at>
    <hosts/>
    <deleted_at>None</deleted_at>
    <id>1</id>
    <metadata>
        <availability_zone>nova</availability_zone>
```

```
</metadata>
</aggregate>
```

### Example 4.126. Remove host from aggregate: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2012-12-04T12:04:26.557909",
    "deleted": false,
    "deleted_at": null,
    "hosts": [],
    "id": 1,
    "metadata": {
      "availability_zone": "nova"
    },
    "name": "name",
    "updated_at": null
  }
}
```

This operation does not return a response body.

## 4.14. Attach interfaces (os-interface)

Create, list, get details for, and delete port interfaces.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/os-interface	Creates and uses a port interface to attach the port to a server instance.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-interface	Lists port interfaces.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}	Shows information about a specified port interface.
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}	Detaches a specified port interface.

## 4.14.1. Create interface

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/os-interface	Creates and uses a port interface to attach the port to a server instance.

Normal response codes: 202

### 4.14.1.1. 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 4.127. Create interface: JSON request

```
{
  "interfaceAttachment": {
    "port_id": "ce531f90-199f-48c0-816c-13e38010b442"
  }
}
```

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

Name	Type	Description
interfaceAttachment	String <i>(Required)</i>	Specify the interfaceAttachment action in the request body.
port_id	Uuid <i>(Required)</i>	The ID of the port for which you want to create an interface.

#### Example 4.128. Create interface: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<interfaceAttachment>
  <port_id>ce531f90-199f-48c0-816c-13e38010b442</port_id>
</interfaceAttachment>
```

This operation does not accept a request body.

### 4.14.1.2. Response

#### Example 4.129. Create interface: JSON response

```
{
  "interfaceAttachment": {
    "fixed_ips": [
      {
        "ip_address": "192.168.1.1",
        "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"
    }
}

```

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

Name	Type	Description
fixed_ipss	String <i>(Required)</i>	Fixed IP addresses with subnet IDs.
mac_addr	String <i>(Required)</i>	The MAC address.
net_id	Uuid <i>(Required)</i>	The network ID.
port_id	Uuid <i>(Required)</i>	The port ID.
port_state	String <i>(Required)</i>	The port state.

### Example 4.130. Create interface: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<interfaceAttachment>
    <net_id>3cb9bc59-5699-4588-a4b1-b87f96708bc6</net_id>
    <port_id>ce531f90-199f-48c0-816c-13e38010b442</port_id>
    <fixed_ipss>
        <fixed_ip>
            <subnet_id>f8a6e8f8-c2ec-497c-9f23-da9616de54ef</subnet_id>
            <ip_address>192.168.1.3</ip_address>
        </fixed_ip>
    </fixed_ipss>
    <port_state>ACTIVE</port_state>
    <mac_addr>fa:16:3e:4c:2c:30</mac_addr>
</interfaceAttachment>

```

This operation does not return a response body.

## 4.14.2. List interfaces

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-interface	Lists port interfaces.

Normal response codes: 202

### 4.14.2.1. 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.

### 4.14.2.2. Response

#### Example 4.131. List interfaces: JSON response

```
{
  "interfaceAttachments": [
    {
      "port_state": "ACTIVE",
      "fixed_ips": [
        {
          "subnet_id": "f8a6e8f8-c2ec-497c-9f23-da9616de54ef",
          "ip_address": "192.168.1.3"
        }
      ],
      "net_id": "3cb9bc59-5699-4588-a4b1-b87f96708bc6",
      "port_id": "ce531f90-199f-48c0-816c-13e38010b442",
      "mac_addr": "fa:16:3e:4c:2c:30"
    }
  ]
}
```

This table shows the body parameters for the list interfaces response:

Name	Type	Description
port_state	String <i>(Required)</i>	The port state.
fixed_ips	String <i>(Required)</i>	Fixed IP addresses with subnet IDs.
mac_addr	String <i>(Required)</i>	The MAC address.
net_id	Uuid <i>(Required)</i>	The network ID.
port_id	Uuid	The port ID.

Name	Type	Description
	(Required)	

### Example 4.132. List interfaces: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<interfaceAttachments>
  <interfaceAttachment>
    <port_state>ACTIVE</port_state>
    <fixed_ips>
      <fixed_ip>
        <subnet_id>f8a6e8f8-c2ec-497c-9f23-da9616de54ef</subnet_id>
        <ip_address>192.168.1.3</ip_address>
      </fixed_ip>
    </fixed_ips>
    <port_id>ce531f90-199f-48c0-816c-13e38010b442</port_id>
    <net_id>3cb9bc59-5699-4588-a4b1-b87f96708bc6</net_id>
    <mac_addr>fa:16:3e:4c:2c:30</mac_addr>
  </interfaceAttachment>
</interfaceAttachments>
```

This operation does not return a response body.

## 4.14.3. Show attached interface information

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}	Shows information about a specified port interface.

**Normal response codes:** 202

### 4.14.3.1. Request

This table shows the URI parameters for the show attached interface information 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.

### 4.14.3.2. Response

#### Example 4.133. Show attached interface information: JSON response

```
{
  "interfaceAttachment": {
    "port_state": "ACTIVE",
    "fixed_ips": [
      {
        "subnet_id": "f8a6e8f8-c2ec-497c-9f23-da9616de54ef",
        "ip_address": "192.168.1.3"
      }
    ],
    "net_id": "3cb9bc59-5699-4588-a4b1-b87f96708bc6",
    "port_id": "ce531f90-199f-48c0-816c-13e38010b442",
    "mac_addr": "fa:16:3e:4c:2c:30"
  }
}
```

#### Example 4.134. Show attached interface information: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<interfaceAttachment>
  <port_state>ACTIVE</port_state>
  <fixed_ips>
    <fixed_ip>
      <subnet_id>b6e47749-6bf0-4d6e-ae4b-ba6b5e238510</subnet_id>
      <ip_address>192.168.123.131</ip_address>
    </fixed_ip>
  </fixed_ips>
  <port_id>89e64f2e-86bd-4c19-9155-4548b36fdcb2</port_id>
  <net_id>a9efd207-2c1a-4cdd-a296-d3c7c3211302</net_id>
  <mac_addr>fa:16:3e:a4:c1:2c</mac_addr>
</interfaceAttachment>
```

This operation does not return a response body.

## 4.14.4. Detach interface

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}	Detaches a specified port interface.

Normal response codes: 202

### 4.14.4.1. 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.

## 4.15. Root certificates (os-certificates)

Create and show details for a root certificate.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/os-certificates	Creates a root certificate.
<b>GET</b>	/v2/{tenant_id}/os-certificates	Shows details for a root certificate owned by a specified tenant ID.

## 4.15.1. Create root certificate

Method	URI	Description
POST	/v2/{tenant_id}/os-certificates	Creates a root certificate.

**Normal response codes:** 202

### 4.15.1.1. Request

This table shows the URI parameters for the create root certificate 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.

### 4.15.1.2. Response

#### Example 4.135. Create root certificate: JSON response

```
{
    "certificate": {
        "data": "Certificate:\n      Data:\n          Version: 1 (0x0)\n          Serial Number: 23 (0x17)\n          Signature Algorithm: md5WithRSAEncryption\n          Issuer: O=NOVA ROOT, L=Mountain View, ST=California, C=US\n          Validity\n              Not Before: Oct 2 19:31:45 2012 GMT\n              Not After : Oct 2 19:31:45 2013 GMT\n              Subject: C=US, ST=California, O=OpenStack, OU=NovaDev, CN=openstack-fake-2012-10-02T19:31:45Z\n              Subject Public Key Info:\n                  Public Key Algorithm: rsaEncryption\n                  RSA Public Key: (1024 bit)\n                      Modulus (1024 bit):\n                          00:b8:87:67:7a:de:28:ed:f6:5d:1f:20:14:58:df:\n                          b0:f7:62:3d:85:61:a8:c2:31:49:5f:b5:2a:07:34:\n                          0e:25:13:0d:2e:4d:79:c7:26:36:dd:d2:3e:c7:36:\n                          54:80:0d:67:80:32:e6:a8:48:33:69:ec:22:2c:5c:\n                          cb:7a:88:0f:c0:48:de:67:14:54:d9:94:b4:6a:23:\n                          36:28:23:44:47:8a:24:89:8e:f4:86:77:59:f8:62:\n                          b6:ab:d5:e0:bc:b4:ad:d1:95:dd:59:a3:aa:e3:ea:\n                          d3:ae:23:17:c5:54:96:a3:25:56:72:90:20:07:8c:\n                          63:4d:be:e9:60:7e:10:57:17\n          Exponent: 65537 (0x10001)\n          Signature Algorithm: md5WithRSAEncryption\n          32:82:ff:8b:92:0e:8d:9c:6b:ce:7e:fe:34:16:2a:4c:47:4f:\n          c7:28:a2:33:1e:48:56:2e:4b:e8:e8:e3:48:b1:3d:a3:43:21:\n          ef:83:e7:df:e2:10:91:7e:9a:c0:4d:1e:96:68:2b:b9:f7:84:\n          7f:ec:84:8a:bf:bc:5e:50:05:d9:ce:4a:1a:bf:d2:bf:0c:d1:\n          7e:ec:64:c3:a5:37:78:a3:a6:2b:a1:b7:1c:cc:c8:b9:78:61:\n          98:50:3c:e6:28:34:f1:0e:62:bb:b5:d7:a1:dd:1f:38:c6:0d:\n          58:9f:81:67:ff:9c:32:fc:52:7e:6d:8c:91:43:49:fe:e3:48:\n          bb:40\n-----BEGIN CERTIFICATE-----\nMIICMzCCAZwCARcwDQYJKoZIhvCNAQEEBQAwtjE5MBAGA1UEChMjTk9WQSBST09U\\nMRYwFAYDVQQHEw1Nb3VudGFpbIBWaWV3MRMwEQYDVQQIEwpDYWxpZm9ybmlhMQsw\\nCQYDVQQGEwJVUzAeFw0xMjEwMDIxOTMxNDVaFw0xMzEwMDIxOTMxNDVaMHYxCzAJ\\nBgNVBAYTA1VTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRIwEAYDVQQKEw1PcGVuU3Rh\\nY2sxEDAOBgNVBAstB05vdmFEZXYxLDAqBgNVBAMTl29wZW5zdGFjay1mYWt1LTlIw\\nMTItMTAtMDJUMTk6MzE6NDVamIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC4\\nh2d63ijt910fIBRY37D3Yj2FYajCMULftSoHNA41Ew0uTXnHjbd0j7HN1SADWeA\\n"
    }
}
```

```

nMuaoSNDp7C1sXMT6iA/ASN5nFFTz1LRqIzYoI0RHiSJjvSGd1n4Yrar1eC8tK3R\
nld1Zo6rj6tOuIxJFVJa jJVZykCAHjGNNvulgfhBXFwIDAQABMA0GCSqGS1b3DQE\B
nBAUAA4GBADKC/4uSDo2ca85+/jQWKkxHT8coojMeSFYuS+jo40ixPaNDIE+D59/i\B
nEJF+msBNHpZoK7n3hH/shIq/vF5QBdnOShq/0r8M0X7sZM01N3ijpiuhtxzMyL14\B
nYZhQPOYoNPPEOYru116HdHzjGDVifgWf/nDL8Un5tjJFDSf7jsLTa\n----END
CERTIFICATE----\n",
"private_key": "----BEGIN RSA PRIVATE KEY----\
nMIICXAIBAAKbQc4h2d63ijt910fIBRY37D3Yj2FYajCMUlftSoHNA41Ew0uTXnH\
nJjbd0j7Hn1SADWeAMuaoSNDp7C1sXMT6iA/ASN5nFFTz1LRqIzYoI0RHiSJjvSG\B
nd1n4Yrar1eC8tK3R1d1Zo6rj6tOuIxJFVJa jJVZykCAHjGNNvulgfhBXFwIDAQAB\B
nAoGBAIjfxx4YU/v01lwUC4OwyS92q3OYcPk6XdakJryZHDTb4NcLmNz jt6bqIK7b\B
n2enyB2fMWdNRWvGiueZ2HmiRLDyOGsAvdEsHvL4qbr9EZGTqC8Qxx+zTevWWf6pB\B
nF1zxzbXNQDFZdf9kVsSLCkbMHITnW1k4MrM++9gfCO3WrfehAkEA4nd8TyCCZazq\B
nKMOQwFLTNaiVLeTXCtvGop14ZNiKYz1qI3KDXb2wbAyArFuERlotxFlylXpwt1Mo\B
nS1I/C/sYqwJBANCX1sdfRJq8DpdP44ThWqOkWFLB9rBiwyBt8746fx8amwr8eyz\B
nH44/z5GT/Vyp8qFs jkuDzeP93eeDnr2qE0UCP1zipRnPO6x4P5J4o+Y+EmLvwkAQ\B
nnCLYAaCvUbILHrbq2Z2wWjEYnEO03RHUD2xjkGH4TgcBMtW4e+ZZEIduwJACnIw\B
nLVfWBbg5QVac3EC021EVoz9xbUnk4Eu2usS4Yrs7USN6QBJQWD1V1cKFg6h3ICJh\B
nleKJ4wsJm9h5kKH9yQJBAN8CaX223M1TSuBOVuIOwNA+09ilfx4UCLiH1fGMKDpe\B
nxVcmkM3qCnTqNxAPSFdT9IyB3IXiaLWbvz17MfiOwQ=\n----END RSA PRIVATE KEY----\n"
}
}

```

#### **Example 4.136. Create root certificate: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<certificate
    private_key="-----BEGIN RSA PRIVATE KEY-----&
#10;MIICXAIBAAKBgQC4h2d63ijt9l0fibRY37D3Yj2FYajCMULftSoHNA4lEw0uTXnH&
#10;Jjb0j7HN1SADWeAMuaoSDNp7C1sXmt6iA/ASN5nFFTz1LrqIzYoI0RHiiSJjvSG&
#10;d1n4YrarleC8tK3Rld1Zo6rj6tOuIxjFVJa jJVZykhCAHjGNNvulgfbXFwIDAQAB&
#10;AoGBAljfxx4YU/v01lwUC4OwyS92q3OYcPk6XdakJryZHDTb4NcLmNzjt6bqIK7b&
#10;2enyB2fMWdNRWvGiueZ2HmiRLDyOGsAVdEsHvL4qbr9EZGTqC8Qxx+zTevWWf6pB&
#10;F1zxzbXNQDFZDf9kVssLCkbMHITnW1k4MrM++9gfCO3WrfehAkeEA4nd8TyCCZazq&
#10;KMOQwFLTNaiVLLeTXCtvGop14ZNiKYZ1qI3KDXb2wbAyArFuERlotxFly1xpwt1Mo&
#10;SLI/C/sYqwJBANCX1sdfrJq8DpdP44ThWqOkWFLB9rBiwyByt8746fX8amwr8eyz&
#10;H44/z5GT/Vyp8qFsjkuDzeP93eeDnr2qE0UCP1zipRnP06x4P5J4o+Y+EmLvwkAQ&
#10;nCLYAaCvUbILhrbqZ22wJyENe003RHUD2xjkGH4TgcBMTmW4e+ZzEIduwJACnIw&
#10;LVfWBbG5QVac3EC021EVoz9XbUnk4Eu2uss4Yrs7USN6QBjQWD1V1cKFg6h3ICJh&
#10;leKJ4wsJm9h5kKH9yQJBAN8CaX223M1TSuBOVuIOwNA+09iLfx4UCLiH1fGMKDpe&
#10;xVcmkM3qCnTqNxAPSfDt9IyB3IXiaLWbvz17MfiOwQ=&#10;-----END RSA PRIVATE
KEY-----&#10;"
```

data="Certificate:&#10;      Data:&#10;      Version: 1 (0x0)&

#10;              Serial Number: 23 (0x17)&#10;      Signature Algorithm:  
md5WithRSAEncryption&#10;      Issuer: O=NOVA ROOT, L=Mountain View,  
ST=California, C=US&#10;      Validity&#10;      Not Before: Oct  
  2 19:31:45 2012 GMT&#10;      Not After : Oct  2 19:31:45 2013  
GMT&#10;              Subject: C=US, ST=California, O=OpenStack, OU=NovaDev,  
CN=openstack-fake-2012-10-02T19:31:45Z&#10;      Subject Public Key  
Info:&#10;              Public Key Algorithm: rsaEncryption&#10;  
    RSA Public Key: (1024 bit)&#10;      Modulus (1024 bit):&

#10;              00:b8:87:67:7a:de:28:ed:f6:5d:1f:20:14:58:df:&  
#10;              b0:f7:62:3d:85:61:a8:c2:31:49:5f:b5:2a:07:34:&  
#10;              0e:25:13:0d:2e:4d:79:c7:26:36:dd:d2:3e:c7:36:&  
#10;              54:80:0d:67:80:32:e6:a8:48:33:69:ec:22:2c:5c:&  
#10;              cb:7a:88:0f:c0:48:de:67:14:54:d9:94:b4:6a:23:&  
#10;              36:28:23:44:47:8a:24:89:8e:f4:86:77:59:f8:62:&  
#10;              b6:ab:d5:e0:bc:b4:ad:d1:95:dd:59:a3:aa:e3:ea:&  
#10;              d3:ae:23:17:c5:54:96:a3:25:56:72:90:20:07:8c:&

```
#10; 63:4d:be:e9:60:7e:10:57:17#10;
Exponent: 65537 (0x10001)&#10; Signature Algorithm: md5WithRSAEncryption&
#10; 32:82:ff:8b:92:0e:8d:9c:6b:ce:7e:fe:34:16:2a:4c:47:4f:&
#10; c7:28:a2:33:1e:48:56:2e:4b:e8:e3:48:b1:3d:a3:43:21:&
#10; ef:83:e7:df:e2:10:91:7e:9a:c0:4d:1e:96:68:2b:b9:f7:84:&
#10; 7f:ec:84:8a:bf:bc:5e:50:05:d9:ce:4a:1a:bf:d2:bf:0c:d1:&
#10; 7e:ec:64:c3:a5:37:78:a3:a6:2b:a1:b7:1c:cc:c8:b9:78:61:&
#10; 98:50:3c:e6:28:34:f1:0e:62:bb:b5:d7:a1:dd:1f:38:c6:0d:&
#10; 58:9f:81:67:ff:9c:32:fc:52:7e:6d:8c:91:43:49:fe:e3:48:&
#10; bb:40#10;-----BEGIN CERTIFICATE-----&
#10;MIICMzCCAZwCARcwDQYJKoZIhvcNAQEEBQAwTjESMBAGA1UEChMJTk9WQSBST09U&
#10;MRYwFAYDVQQHEw1Nb3VudGFpbIBWaWV3MRMwEQYDVQQIEwpDYWxpZm9ybmlhMQsw&
#10;CQYDVQQGEwJVUzAeFw0xMjEwMDIxOTMxNDVaFw0xMzEwMDIxOTMxNDVaMHYxCzAJ&
#10;BgNVBAYTA1VTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRIwEAYDVQQKEwlPcGVuU3Rh&
#10;Y2sxEDAObgNVBAsTB05vdmFEZXYxLDAqBgNVBAMTI29wZW5zdGFjay1mYWt1LTIw&
#10;MTItMTAtMDJUMTk6MzE6NDVaMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC4&
#10;h2d63ijt910fIBRY37D3Yj2FYajCMUlftSoHNA41Ew0uTXnHJjbd0j7HN1SADWeA&
#10;MuaoSDNp7CIsXMt6iA/ASN5nFFTZ1LRqIzYoI0RHiiSJjvSGd1n4Yrar1eC8tK3R&
#10;ld1Zo6rj6tOuIxFFVJa jJVZykCAHjGNNvulgfhBXFwIDAQABMA0GCSqGSIb3DQEBA&
#10;BAUAA4GBADKC/4uSDo2ca85+/jQWKxHT8coojMeSFYU S+jo40ixPaNDIe+D59/i&
#10;EJF+msBNHpZoK7n3hH/shIq/vF5QBdnOShq/0r8M0X7sZMO1N3ijpiuhtxzMyL14&
#10;YZhQPOYOONPEOYru116HzjGDVifgWf/nDL8Un5tjJFDSf7jsLtA#10;-----END
CERTIFICATE-----#10; "/>
```

This operation does not return a response body.

## 4.15.2. Show root certificate details

Method	URI	Description
GET	/v2/{tenant_id}/os-certificates	Shows details for a root certificate owned by a specified tenant ID.

**Normal response codes:** 200

### 4.15.2.1. Request

This table shows the URI parameters for the show root certificate details 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.

### 4.15.2.2. Response

#### Example 4.137. Show root certificate details: JSON response

```
{
  "certificate": {
    "data": "-----BEGIN CERTIFICATE-----\\nMIICyzCCAjSgAwIBAgIJAIJ/UoFWKoOUMA0GCSqGSIB3DQEBAUAME4xEjAQBgNV\\nbAoTCU5PVkEgUk9PVDEWMBQGA1UEBxMNTW91bnRhaW4gVm1ldzETMBEGA1UECBMK\\nQ2FsaWZvcm5pYTELMAkGA1UEBhMCVVMwHhcNMTIxMDAyMTg1NzQ1WhcNMTMxMDAy\\nMTg1NzQ1WjBOMRIweAYDVQQKEw1OT1ZBIFJPT1QxFjaUBgNVBACTDU1vdW50YWlu\\nIFZpZXcxEzARBgNVBAgTCKNhbg1mb3JuaWEExCzAJBgNVBAYTA1VTMIGfMA0GCSqG\\nSIB3DQEBAQUAA4GNADCBiQKBgQCg0Bn8WSqbJF3QNTZUxo1TzmFBxuqvjhZLKbnQ\\nIiShdVIWUK7RC8frq8FJI7dgJNmvkIBn9njABWD0zmurQRCzD65yCSbUc4R2ea5H\\nIK4wQIui0CJykvmBNjAe3bzttVVsvs/ccDTs jtqq3F/KeQkKzQVfSWBrJSmYtG5tO\\nG+dOSwIDAQABo4GwMIGtMAwGA1UdEwQFMAMBAf8wHQYDVR0OBBYEFC1jRfaNOsA/\\n9mHuq0io7Lt83FtaMH4GA1UdIwR3MHWAFCl jRfaNOsA/9mHuq0io7Lt83FtaoVKk\\nUDBOMRIweAYDVQQKEw1OT1ZBIFJPT1QxFjaUBgNVBACTDU1vdW50YWluIFZpZXcx\\neZARBgNVBAgTCKNhbg1mb3JuaWEExCzAJBgNVBAYTA1VTggkAgn9SgVYqg5QwDQYJ\\nKoZIhvcaNAQEEBQADgYEAEbpJOOlpcKCh5omwfAwAfFg1ml4h/FJiCH3PETmOCc+31\\nCTWTBd4MG8AoH7A3PU2JKAGVQ5XWo6+ihpW1RgfQpCnloI6vIeGcws+rSLnlzULT\\nIvfCJpRg7iQdR3jZGt3295behtP1GsCqipJEulOkOaEis8iLlXgSOG94Mkwlb4Q=\n-----END CERTIFICATE-----\\n",
    "private_key": null
  }
}
```

#### Example 4.138. Show root certificate details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<certificate private_key="None">
  <data>-----BEGIN CERTIFICATE-----#10;MIICyzCCAjSgAwIBAgIJAIJ/UoFWKoOUMA0GCSqGSIB3DQEBAUAME4xEjAQBgNV#10;BAoTCU5PVkEgUk9PVDEWMBQGA1UEBxMNTW91bnRhaW4gVm1ldzETMBEGA1UECBMK#10;#10;Q2FsaWZvcm5pYTELMAkGA1UEBhMCVVMwHhcNMTIxMDAyMTg1NzQ1WhcNMTMxMDAy#10;#10;MTg1NzQ1WjBOMRIweAYDVQQKEw1OT1ZBIFJPT1QxFjaUBgNVBACTDU1vdW50YWlu#10;#10;IFZpZXcxEzARBgNVBAgTCKNhbg1mb3JuaWEExCzAJBgNVBAYTA1VTMIGfMA0GCSqG#10;#10;SIB3DQEBAQUAA4GNADCBiQKBgQCg0Bn8WSqbJF3QNTZUxo1TzmFBxuqvjhZLKbnQ&
```

```
#10 ; II ShdVIWUK7RC8frq8FJI7dgJNmVkBn9njABWDoZmurQRCzD65yCSbUc4R2ea5H&
#10 ; IK4wQIui0CJykvMBNjAe3bzrVV8/ccDTsjtqq3F/KeQkKzQVfSWBrJSmYtG5tO&
#10 ; G+dOSwIDAQABo4GwMIGtMAwGA1UdEwQFMAMBAf8wHQYDVR0OBByEFC1jRfaNOsA/&
#10 ; 9mHuq0io7Lt83FtaMH4GA1UdIwR3MHWAFCljRfaNOsA/9mHuq0io7Lt83FtaoVKk&
#10 ; UDBOMRIwEAYDVQQKEw1OT1ZBIFJPT1QxFjAUBgNVBActDU1vdW50YWluIFZpZXcx&
#10 ; EzARBgNVBAgTCkNhbGlmb3JuaWExCzAJBgNVBAYTA1VTggkAgn9SgVYqg5QwDQYJ&
#10 ; KoZIhvcNAQEEBQADgYEAEbpJOOlPCh5omwfAwAfFg1ml4h/FJiCH3PETmOCc+31&
#10 ; CtWTBd4MG8AoH7A3PU2JKAGVQ5XWc6+ihpW1RgfQpCnloI6vIeGcws+rSLnlzULT&
#10 ; IvfCJpRg7iQdR3jZGt3295behtP1GsCqipJEulOkOaEIs8iL1XgSOG94Mkwlb4Q=&#10 ; ----
END CERTIFICATE----&#10 ; "/>
```

This operation does not return a response body.

## 4.16. Cloudpipe (os-cloudpipe)

Manage virtual VPNs for projects.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-cloudpipe	Lists cloudpipes.
<b>POST</b>	/v2/{tenant_id}/os-cloudpipe{?project_id}	Creates a cloudpipe.
<b>POST</b>	/v2/{tenant_id}/os-cloudpipe/configure-project	Updates the virtual private network (VPN) IP address and port for a specified cloudpipe instance.

## 4.16.1. List cloudpipes

Method	URI	Description
GET	/v2/{tenant_id}/os-clouddpipe	Lists cloudpipes.

**Normal response codes:** 200

### 4.16.1.1. 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.

### 4.16.1.2. Response

#### Example 4.139. List cloudpipes: JSON response

```
{
  "clouddpipes": [
    {
      "created_at": "2012-11-27T17:18:01Z",
      "instance_id": "27deecdb-baa3-4a26-9c82-32994b815b01",
      "internal_ip": "192.168.0.3",
      "project_id": "clouddpipe-fa1765bd-a352-49c7-a6b7-8ee108a3cb0c",
      "public_ip": "127.0.0.1",
      "public_port": 22,
      "state": "down"
    }
  ]
}
```

#### Example 4.140. List cloudpipes: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<clouddpipes>
  <clouddpipe>
    <public_ip>127.0.0.1</public_ip>
    <created_at>2012-09-25T18:18:49Z</created_at>
    <public_port>22</public_port>
    <state>down</state>
    <instance_id>1a17f615-343b-430f-976a-457c029eddb7</instance_id>
    <internal_ip>192.168.0.3</internal_ip>
    <project_id>clouddpipe-6405f2ca-caf9-493b-a1f6-e55f595d75ab</project_id>
  </clouddpipe>
</clouddpipes>
```

This operation does not return a response body.

## 4.16.2. Create cloupipe

Method	URI	Description
POST	/v2/{tenant_id}/os-cloupipe{?project_id}	Creates a cloupipe.

Normal response codes: 200

### 4.16.2.1. 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.

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

Name	Type	Description
project_id	String <i>(Optional)</i>	Creates the cloupipe for the specified project ID. If omitted, the project ID defaults to the calling tenant.

#### Example 4.141. Create cloupipe: JSON request

```
{
  "cloupipe": {
    "project_id": "cloupipe-059f21e3-c20e-4efc-9e7a-eba2ab3c6f9a"
  }
}
```

#### Example 4.142. Create cloupipe: XML request

```
<cloupipe
  project_id="cloupipe-6405f2ca-caf9-493b-a1f6-e55f595d75ab"
/>
```

This operation does not accept a request body.

### 4.16.2.2. Response

#### Example 4.143. Create cloupipe: JSON response

```
{
  "instance_id": "1e9b8425-34af-488e-b969-4d46f4a6382e"
}
```

#### Example 4.144. Create cloupipe: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<cloupipe>
  <instance_id>1a17f615-343b-430f-976a-457c029eddb7</instance_id>
</cloupipe>
```

This operation does not return a response body.

## 4.16.3. Update cloupipe

Method	URI	Description
POST	/v2/{tenant_id}/os-cloupipe/configure-project	Updates the virtual private network (VPN) IP address and port for a specified cloupipe instance.

Normal response codes: 202

### 4.16.3.1. Request

This table shows the header parameters for the update cloupipe request:

Name	Type	Description
vpn_ip	String <i>(Required)</i>	The VPN IP address.
vpn_port	String <i>(Required)</i>	The VPN port.

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 4.145. Update cloupipe: JSON request

```
{
  "configure_project": {
    "vpn_ip": "192.168.1.1",
    "vpn_port": "2000"
  }
}
```

### Example 4.146. Update cloupipe: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<configure_project>
  <vpn_ip>192.168.1.1</vpn_ip>
  <vpn_port>2000</vpn_port>
</configure_project>
```

This operation does not accept a request body.

## 4.17. Server console output (os-console-output)

Get console output for a server instance.

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

## 4.17.1. Get console output for an instance

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

Normal response codes: 200

### 4.17.1.1. Request

This table shows the URI parameters for the get console output for an instance 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 4.147. Get console output: JSON request

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

This table shows the body parameters for the get console output for an instance request:

Name	Type	Description
os-getConsoleOutput	String <i>(Required)</i>	Specify the os-getConsoleOutput action in the request body.
length	<i>(Optional)</i>	Number of lines to fetch from the end of console log.

#### Example 4.148. Get console output: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<os-getConsoleOutput length="50" />
```

This operation does not accept a request body.

### 4.17.1.2. Response

#### Example 4.149. Get console output: JSON response

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

#### Example 4.150. Get console output: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<output>FAKE CONSOLE OUTPUT
ANOTHER
LAST LINE</output>
```

This operation does not return a response body.

## 4.18. Server console (os-consoles)

Get a console for a server instance.

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

## 4.18.1. Get console

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

**Normal response codes:** 200

### 4.18.1.1. Request

This table shows the URI parameters for the get console 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 4.151. Get console: JSON request

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

This table shows the body parameters for the get console request:

Name	Type	Description
getConsole	String <i>(Required)</i>	Specify the getConsole action in the request body.
type	String <i>(Required)</i>	Valid values are novnc and xvpxvnc.

#### Example 4.152. Get console: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<os>getVNCConsole type="novnc" />
```

This operation does not accept a request body.

### 4.18.1.2. Response

#### Example 4.153. Get console: JSON response

```
{
  "console": {
    "type": "novnc",
    "url": "http://example.com:6080/vnc_auto.html?token=f9906a48-
b71e-4f18-baca-c987da3ebdb3&title=dafa(75eceef58-3b8e-4659-ab3b-5501454188e9)"
  }
}
```

**Example 4.154. Get console: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<console>
  <type>novnc</type>
  <url>http://example.com:6080/vnc_auto.html?token=f9906a48-b71e-4f18-baca-
c987da3ebdb3</url>
</console>
```

This operation does not return a response body.

## 4.19. Coverage reports (os-coverage)

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/os-coverage/action	Generates a coverage report.
<b>POST</b>	/v2/{tenant_id}/os-coverage/action	Starts Nova coverage reporting.
<b>POST</b>	/v2/{tenant_id}/os-coverage/action	Starts coverage reporting for all services.
<b>POST</b>	/v2/{tenant_id}/os-coverage/action	Stops coverage reporting.

## 4.19.1. Get coverage report

Method	URI	Description
POST	/v2/{tenant_id}/os-coverage/action	Generates a coverage report.

Normal response codes: 200

### 4.19.1.1. Request

This table shows the URI parameters for the get coverage report request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.155. Start combined report: JSON request

```
{
  "report": {
    "xml": true,
    "file": "report"
  }
}
```

This table shows the body parameters for the get coverage report request:

Name	Type	Description
file	String <i>(Required)</i>	The file name for the report.
xml	Bool <i>(Optional)</i>	Set to true to generate the report in XML format.
html	Bool <i>(Optional)</i>	Set to true to generate the report in HTML format.

#### Example 4.156. Start combined report: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<report>
  <file>report</file>
  <xml>True</xml>
</report>
```

This operation does not accept a request body.

### 4.19.1.2. Response

#### Example 4.157. Get coverage report: JSON response

```
{
  "path": "/tmp/tmp6kdYaa/nova-coverage_TOTUbz/report"
}
```

This table shows the body parameters for the get coverage report response:

Name	Type	Description
path	String <i>(Required)</i>	The path to the report file.

**Example 4.158. Get coverage report: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<path>/tmp/tmp4j87bp/nova-coverage_7ViTA7/report</path>
```

This operation does not return a response body.

## 4.19.2. Start coverage report

Method	URI	Description
POST	/v2/{tenant_id}/os-coverage/action	Starts Nova coverage reporting.

**Normal response codes:** 202

### 4.19.2.1. Request

This table shows the URI parameters for the start coverage report request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.159. Start coverage report: JSON request

```
{
    "start": {}
}
```

This table shows the body parameters for the start coverage report request:

Name	Type	Description
start	String <i>(Required)</i>	Specify start in the request body.

#### Example 4.160. Start coverage report: XML request

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

This operation does not accept a request body.

## 4.19.3. Start combined coverage report

Method	URI	Description
POST	/v2/{tenant_id}/os-coverage/action	Starts coverage reporting for all services.

All reports are combined into a single report.

**Normal response codes:** 200

### 4.19.3.1. Request

This table shows the URI parameters for the start combined coverage report request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.161. Start combined report: JSON request

```
{
  "start": {
    "combine": true
  }
}
```

This table shows the body parameters for the start combined coverage report request:

Name	Type	Description
start	String <i>(Required)</i>	Specify start in the request body.
combine	Bool <i>(Required)</i>	Set to true to generate a combined report for all services.

#### Example 4.162. Start combined report: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<start>
  <combine>True</combine>
</start>
```

This operation does not accept a request body.

## 4.19.4. Stop coverage report

Method	URI	Description
POST	/v2/{tenant_id}/os-coverage/action	Stops coverage reporting.

**Normal response codes:** 202

### 4.19.4.1. Request

This table shows the URI parameters for the stop coverage report request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.163. Stop coverage report: JSON request

```
{
    "stop": {}
}
```

This table shows the body parameters for the stop coverage report request:

Name	Type	Description
stop	String <i>(Required)</i>	Specify stop in the request body.

#### Example 4.164. Stop coverage report: XML request

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

This operation does not accept a request body.

### 4.19.4.2. Response

#### Example 4.165. Stop report: JSON response

```
{
    "path": "/tmp/tmpua9HvB/nova-coverage_rs2CaS"
```

#### Example 4.166. Stop report: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<path>/tmp/tmpCLve38/nova-coverage_GJ4BZ_</path>
```

This operation does not return a response body.

## 4.20. Server deferred delete (os-deferred-delete)

Force-delete a server or restore a deleted server.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Force-deletes a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Restores a deleted server.

## 4.20.1. Force delete server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Force-deletes a server.

Normal response codes: 202

### 4.20.1.1. Request

This table shows the URI parameters for the force delete server 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 4.167. Force delete server: JSON request

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

This table shows the body parameters for the force delete server request:

Name	Type	Description
forceDelete	String <i>(Required)</i>	Specify the forceDelete action in the request body.

#### Example 4.168. Force delete server: XML request

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

This operation does not accept a request body.

## 4.20.2. Restore server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Restores a deleted server.

Normal response codes: 202

### 4.20.2.1. Request

This table shows the URI parameters for the restore server 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 4.169. Restore server: JSON request

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

This table shows the body parameters for the restore server request:

Name	Type	Description
restore	String <i>(Required)</i>	Specify the <code>restore</code> action in the request body.

#### Example 4.170. Restore server: XML request

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

This operation does not accept a request body.

## 4.21. Fixed IPs (os-fixed-ips)

Show data for a specified fixed IP, such as host name, CIDR, and address. Also, reserve or free a fixed IP.

Method	URI	Description
GET	/v2/{tenant_id}/os-fixed-ips/{fixed_ip}	Shows information for a specified fixed IP address.
POST	/v2/{tenant_id}/os-fixed-ips/{fixed_ip}/action	Reserves or releases a fixed IP.

## 4.21.1. Show fixed IP information

Method	URI	Description
GET	/v2/{tenant_id}/os-fixed-ips/{fixed_ip}	Shows information for a specified fixed IP address.

**Normal response codes:** 200

### 4.21.1.1. Request

This table shows the URI parameters for the show fixed ip information 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.

### 4.21.1.2. Response

#### Example 4.171. Show fixed IP information: JSON response

```
{
  "fixed_ip": {
    "address": "192.168.1.1",
    "cidr": "192.168.1.0/24",
    "host": "host",
    "hostname": "openstack"
  }
}
```

#### Example 4.172. Show fixed IP information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<fixed_ip>
  <cidr>192.168.1.0/24</cidr>
  <hostname>openstack</hostname>
  <host>host</host>
  <address>192.168.1.1</address>
</fixed_ip>
```

This operation does not return a response body.

## 4.21.2. Reserve or release a fixed IP

Method	URI	Description
<b>POST</b>	/v2/{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

### 4.21.2.1. 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 4.173. Reserve or release a fixed IP: JSON request

```
{
    "reserve": "None"
}
```

#### Example 4.174. Reserve or release a fixed IP: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<reserve>None</reserve>
```

This operation does not accept a request body.

## 4.22. Floating IP DNS records (os-floating-ip-dns)

Manage 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
<b>GET</b>	/v2/{tenant_id}/os-floating-ip-dns	Lists registered DNS domains published by the DNS drivers.
<b>PUT</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}	Creates or updates a DNS domain.
<b>DELETE</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}	Deletes a DNS domain and all associated host entries.
<b>PUT</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Creates or updates a DNS entry.
<b>GET</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Finds a unique DNS entry for a specified domain and name.
<b>DELETE</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Deletes a specified DNS entry.
<b>GET</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}	Lists DNS entries for a specified domain and IP.

## 4.22.1. List DNS domains

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ip-dns	Lists registered DNS domains published by the DNS drivers.

**Normal response codes:** 200

### 4.22.1.1. Request

This table shows the URI parameters for the list dns domains request:

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

This operation does not accept a request body.

### 4.22.1.2. Response

#### Example 4.175. List DNS domains: JSON response

```
{
  "domain_entries": [
    {
      "availability_zone": null,
      "domain": "domain1.example.org",
      "project": "project1",
      "scope": "public"
    }
  ]
}
```

#### Example 4.176. List DNS domains: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<domain_entries>
  <domain_entry project="project1" scope="public" domain="domain1.example.org" availability_zone="None"/>
</domain_entries>
```

This operation does not return a response body.

## 4.22.2. Create or update DNS domain

Method	URI	Description
PUT	/v2/{tenant_id}/os-floating-ip-dns/{domain}	Creates or updates a DNS domain.

**Normal response codes:** 200

### 4.22.2.1. Request

This table shows the URI parameters for the create or update dns domain request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.

#### Example 4.177. Create or update DNS domain: JSON request

```
{
  "domain_entry": {
    "domain": "domain1.example.org",
    "scope": "public",
    "project": "project1"
  }
}
```

#### Example 4.178. Create or update DNS domain: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<domain_entry>
  <domain>domain1.example.org</domain>
  <scope>public</scope>
  <project>project1</project>
</domain_entry>
```

This operation does not accept a request body.

### 4.22.2.2. Response

#### Example 4.179. Create or update DNS domain: JSON response

```
{
  "domain_entry": {
    "availability_zone": null,
    "domain": "domain1.example.org",
    "project": "project1",
    "scope": "public"
  }
}
```

#### Example 4.180. Create or update DNS domain: XML response

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

```
<domain_entry project="project1" scope="public" domain="domain1.example.org"
availability_zone="None" />
```

This operation does not return a response body.

## 4.22.3. Delete DNS domain

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}	Deletes a DNS domain and all associated host entries.

**Normal response codes:** 200

### 4.22.3.1. Request

This table shows the URI parameters for the delete dns domain request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.

This operation does not accept a request body.

## 4.22.4. Create or update DNS entry

Method	URI	Description
PUT	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Creates or updates a DNS entry.

Normal response codes: 200

### 4.22.4.1. Request

This table shows the URI parameters for the create or update dns entry request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.
{name}	String	DNS entry name assigned under a domain.

#### Example 4.181. Create or update DNS entry: JSON request

```
{
  "dns_entry": {
    "ip": "192.168.53.11",
    "dns_type": "A"
  }
}
```

#### Example 4.182. Create or update DNS entry: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<dns_entry>
  <ip>192.168.53.11</ip>
  <dns_type>A</dns_type>
</dns_entry>
```

This operation does not accept a request body.

### 4.22.4.2. Response

#### Example 4.183. Create or update DNS entry: JSON response

```
{
  "dns_entry": {
    "domain": "domain1.example.org",
    "id": null,
    "ip": "192.168.1.1",
    "name": "instance1",
    "type": "A"
  }
}
```

#### Example 4.184. Create or update DNS entry: XML response

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

```
<dns_entry ip="192.168.1.1" domain="domain1.example.org" type="A" id="None"
name="instance1" />
```

This operation does not return a response body.

## 4.22.5. Find unique DNS entry

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Finds a unique DNS entry for a specified domain and name.

**Normal response codes:** 200

### 4.22.5.1. Request

This table shows the URI parameters for the find unique dns entry request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.
{name}	String	DNS entry name assigned under a domain.

This operation does not accept a request body.

### 4.22.5.2. Response

#### Example 4.185. Find unique DNS entry: JSON response

```
{
  "dns_entry": {
    "domain": "domain1.example.org",
    "id": null,
    "ip": "192.168.1.1",
    "name": "instance1",
    "type": null
  }
}
```

#### Example 4.186. Find unique DNS entry: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<dns_entry ip="192.168.1.1" domain="domain1.example.org" type="None" id="None" name="instance1"/>
```

This operation does not return a response body.

## 4.22.6. Delete DNS entry

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Deletes a specified DNS entry.

**Normal response codes:** 200

### 4.22.6.1. Request

This table shows the URI parameters for the delete dns entry request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.
{name}	String	DNS entry name assigned under a domain.

This operation does not accept a request body.

## 4.22.7. List DNS entries

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}	Lists DNS entries for a specified domain and IP.

Normal response codes: 200

### 4.22.7.1. Request

This table shows the URI parameters for the list dns entries request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.

This operation does not accept a request body.

### 4.22.7.2. Response

#### Example 4.187. List DNS entries: JSON response

```
{
  "dns_entries": [
    {
      "domain": "domain1.example.org",
      "id": null,
      "ip": "192.168.1.1",
      "name": "instance1",
      "type": null
    }
  ]
}
```

#### Example 4.188. List DNS entries: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<dns_entries>
  <dns_entry ip="192.168.1.1" domain="domain1.example.org" type="None" id="None" name="instance1"/>
</dns_entries>
```

This operation does not return a response body.

## 4.23. Floating IP pools (os-floating-ip-pools)

Manage groups of floating IPs.

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ip-pools	Lists floating IP pools.

## 4.23.1. List floating IP pools

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-floating-ip-pools	Lists floating IP pools.

Normal response codes: 200

### 4.23.1.1. Request

This table shows the URI parameters for the list floating ip pools request:

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

This operation does not accept a request body.

### 4.23.1.2. Response

#### Example 4.189. List floating IP pools: JSON response

```
{
  "floating_ip_pools": [
    {
      "name": "pool1"
    },
    {
      "name": "pool2"
    }
  ]
}
```

#### Example 4.190. List floating IP pools: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ip_pools>
  <floating_ip_pool name="pool1"/>
  <floating_ip_pool name="pool2"/>
</floating_ip_pools>
```

This operation does not return a response body.

## 4.24. Floating IPs (os-floating-ips)

Assign and allocate floating IP addresses to instances that run in an OpenStack cloud.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-floating-ips	Lists floating IP addresses associated with the tenant or account.
<b>POST</b>	/v2/{tenant_id}/os-floating-ips	Allocates a new floating IP address to a tenant or account.
<b>GET</b>	/v2/{tenant_id}/os-floating-ips/{id}	Shows information for a specified floating IP address.

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/os-floating-ips/{id}	Deallocates the floating IP address associated with floating_IP_address_ID.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to an instance.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Removes a floating IP from an instance.

## 4.24.1. List floating IPs

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ips	Lists floating IP addresses associated with the tenant or account.

**Normal response codes:** 200

### 4.24.1.1. Request

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

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

This operation does not accept a request body.

### 4.24.1.2. Response

#### Example 4.191. 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"
    }
  ]
}
```

#### Example 4.192. List floating IPs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ips>
  <floating_ip instance_id="None" ip="10.10.10.1" fixed_ip="None" id="1" pool="nova"/>
  <floating_ip instance_id="None" ip="10.10.10.2" fixed_ip="None" id="2" pool="nova"/>
</floating_ips>
```

This operation does not return a response body.

## 4.24.2. Allocate floating IP

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/os-floating-ips	Allocates a new floating IP address to a tenant or account.

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.

### 4.24.2.1. Request

This table shows the URI parameters for the allocate floating ip request:

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

#### Example 4.193. Allocate floating IP: JSON request

```
{
    "pool": "nova"
}
```

This table shows the body parameters for the allocate floating ip request:

Name	Type	Description
pool	(Optional)	Pool to allocate IP address from. Will use default pool if not specified.

#### Example 4.194. Allocate floating IP: XML request

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

This operation does not accept a request body.

### 4.24.2.2. Response

#### Example 4.195. Allocate floating IP: JSON response

```
{
    "floating_ip": {
        "fixed_ip": null,
        "id": 1,
        "instance_id": null,
        "ip": "10.10.10.1",
        "pool": "nova"
    }
}
```

**Example 4.196. Allocate floating IP: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ip instance_id="None" ip="10.10.10.1" fixed_ip="None" id="1" pool=
"nova"/>
```

This operation does not return a response body.

## 4.24.3. Show floating IP information

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ips/{id}	Shows information for a specified floating IP address.

**Normal response codes:** 200

### 4.24.3.1. Request

This table shows the URI parameters for the show floating ip information request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	String	The unique identifier associated with allocated floating IP address.

This operation does not accept a request body.

### 4.24.3.2. Response

#### Example 4.197. Show floating IP information: JSON response

```
{
  "floating_ip": {
    "fixed_ip": null,
    "id": 1,
    "instance_id": null,
    "ip": "10.10.10.1",
    "pool": "nova"
  }
}
```

#### Example 4.198. Show floating IP information: XML response

```
<floating_ip instance_id="None" ip="10.10.10.1" fixed_ip="None" id="1" pool="nova" />
```

This operation does not return a response body.

## 4.24.4. Deallocate floating IP

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/os-floating-ips/{id}	Deallocates the floating IP address associated with floating_IP_address_ID.

**Normal response codes:** 202

### 4.24.4.1. Request

This table shows the URI parameters for the deallocate floating ip request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	String	The unique identifier associated with allocated floating IP address.

This operation does not accept a request body.

## 4.24.5. Add floating IP

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to an instance.

Normal response codes: 202

### 4.24.5.1. Request

This table shows the URI parameters for the add floating ip request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{server_id}	UUID	The UUID for the server of interest to you.

#### Example 4.199. Add floating IP: JSON request

```
{
  "addFloatingIp": {
    "address": "10.10.10.1"
  }
}
```

This table shows the body parameters for the add floating ip request:

Name	Type	Description
pool	String <i>(Required)</i>	The pool to which to add the floating IP address.

#### Example 4.200. Add floating IP: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<addFloatingIp>
  <address>10.10.10.1</address>
</addFloatingIp>
```

This operation does not accept a request body.

## 4.24.6. Remove floating IP

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Removes a floating IP from an instance.

Normal response codes: 202

### 4.24.6.1. Request

This table shows the URI parameters for the remove floating ip request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{server_id}	UUID	The UUID for the server of interest to you.

#### Example 4.201. Remove floating IP: JSON request

```
{
  "removeFloatingIp": {
    "address": "10.10.10.1"
  }
}
```

This table shows the body parameters for the remove floating ip request:

Name	Type	Description
removeFloatingIp	String <i>(Required)</i>	Specify the removeFloatingIp action in the request body.
address	String <i>(Required)</i>	The floating IP address that you want to remove.

#### Example 4.202. Remove floating IP: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<removeFloatingIp>
  <address>10.10.10.1</address>
</removeFloatingIp>
```

This operation does not accept a request body.

## 4.25. Floating IPs bulk (os-floating-ips-bulk)

Bulk-create, delete, and list floating IPs. By default, the pool is named nova. Use the os-floating-ip-pools extension to view available pools.

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ips-bulk	Lists all floating IPs.
POST	/v2/{tenant_id}/os-floating-ips-bulk	Bulk-creates floating IPs.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/os-floating-ips-bulk/delete	Bulk-deletes floating IPs.
<b>GET</b>	/v2/{tenant_id}/os-floating-ips-bulk/{host_name}	Lists all floating IPs for a specified host.

## 4.25.1. List floating IPs

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ips-bulk	Lists all floating IPs.

**Normal response codes:** 200

### 4.25.1.1. Request

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

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

This operation does not accept a request body.

### 4.25.1.2. Response

#### Example 4.203. List floating IPs: JSON response

```
{
  "floating_ip_info": [
    {
      "address": "10.10.10.1",
      "instance_uuid": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    },
    {
      "address": "10.10.10.2",
      "instance_uuid": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    },
    {
      "address": "10.10.10.3",
      "instance_uuid": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    }
  ]
}
```

#### Example 4.204. List floating IPs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ip_info>
  <item>
    <interface>eth0</interface>
    <instance_uuid>None</instance_uuid>
    <project_id>None</project_id>
```

```
<pool>nova</pool>
<address>10.10.10.1</address>
</item>
<item>
<interface>eth0</interface>
<instance_uuid>None</instance_uuid>
<project_id>None</project_id>
<pool>nova</pool>
<address>10.10.10.2</address>
</item>
<item>
<interface>eth0</interface>
<instance_uuid>None</instance_uuid>
<project_id>None</project_id>
<pool>nova</pool>
<address>10.10.10.3</address>
</item>
</floating_ip_info>
```

This operation does not return a response body.

## 4.25.2. Create floating IPs

Method	URI	Description
POST	/v2/{tenant_id}/os-floating-ips-bulk	Bulk-creates floating IPs.

Normal response codes: 200

### 4.25.2.1. Request

This table shows the URI parameters for the create floating ips request:

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

#### Example 4.205. Create floating IPs: JSON request

```
{
    "floating_ips_bulk_create": {
        "ip_range": "192.168.1.0/24",
        "pool": "nova",
        "interface": "eth0"
    }
}
```

This table shows the body parameters for the create floating ips request:

Name	Type	Description
ip_range	Int <i>(Required)</i>	The range of IP addresses to use for creating floating IPs.

#### Example 4.206. Create floating IPs: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ips_bulk_create>
<ip_range>192.168.1.0/24</ip_range>
<pool>nova</pool>
<interface>eth0</interface>
</floating_ips_bulk_create>
```

This operation does not accept a request body.

### 4.25.2.2. Response

#### Example 4.207. Create floating IPs: JSON response

```
{
    "floating_ips_bulk_create": {
        "interface": "eth0",
        "ip_range": "192.168.1.0/24",
        "pool": "nova"
    }
}
```

**Example 4.208. Create floating IPs: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ips_bulk_create>
  <interface>eth0</interface>
  <ip_range>192.168.1.0/24</ip_range>
  <pool>nova</pool>
</floating_ips_bulk_create>
```

This operation does not return a response body.

## 4.25.3. Bulk-delete floating IPs

Method	URI	Description
POST	/v2/{tenant_id}/os-floating-ips-bulk/delete	Bulk-deletes floating IPs.

Normal response codes: 200

### 4.25.3.1. Request

This table shows the URI parameters for the bulk-delete floating ips request:

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

#### Example 4.209. Bulk-delete floating IPs: JSON request

```
{
    "ip_range": "192.168.1.0/24"
}
```

This table shows the body parameters for the bulk-delete floating ips request:

Name	Type	Description
ip_range	Int <i>(Required)</i>	The range of IP addresses from which to bulk-delete floating IPs.

#### Example 4.210. Bulk-delete floating IPs: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<ip_range>192.168.1.0/24</ip_range>
```

This operation does not accept a request body.

### 4.25.3.2. Response

#### Example 4.211. Bulk-delete floating IPs: JSON response

```
{
    "floating_ips_bulk_delete": "192.168.1.0/24"
}
```

#### Example 4.212. Bulk-delete floating IPs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ips_bulk_delete>192.168.1.0/24</floating_ips_bulk_delete>
```

This operation does not return a response body.

## 4.25.4. List floating IPs by host

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ips-bulk/{host_name}	Lists all floating IPs for a specified host.

**Normal response codes:** 200

### 4.25.4.1. Request

This table shows the URI parameters for the list floating ips by host request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{host_name}	String	The name of the host of interest to you.

This operation does not accept a request body.

### 4.25.4.2. Response

#### Example 4.213. List floating IPs by host: JSON response

```
{
  "floating_ip_info": [
    {
      "address": "10.10.10.3",
      "instance_uuid": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    }
  ]
}
```

#### Example 4.214. List floating IPs by host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ip_info>
  <item>
    <interface>eth0</interface>
    <instance_uuid>None</instance_uuid>
    <project_id>None</project_id>
    <pool>nova</pool>
    <address>10.10.10.3</address>
  </item>
</floating_ip_info>
```

This operation does not return a response body.

## 4.26. Hosts (os-hosts)

Manage physical hosts.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-hosts{?service,zone}	Lists hosts.
<b>GET</b>	/v2/{tenant_id}/os-hosts/{host_name}	Shows information for a specified host.
<b>PUT</b>	/v2/{tenant_id}/os-hosts/{host_name}	Enables a host or puts it in maintenance mode.
<b>GET</b>	/v2/{tenant_id}/os-hosts/{host_name}/startup	Starts a host.
<b>GET</b>	/v2/{tenant_id}/os-hosts/{host_name}/shutdown	Shuts down a host.
<b>GET</b>	/v2/{tenant_id}/os-hosts/{host_name}/reboot	Reboots a host.

## 4.26.1. List hosts

Method	URI	Description
GET	/v2/{tenant_id}/os-hosts{?service,zone}	Lists hosts.

Normal response codes: 200

### 4.26.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

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

Name	Type	Description
service	String <i>(Optional)</i>	The service name. To filter the returned list by service name, specify ?service={service} in the URI.
zone	String <i>(Optional)</i>	The zone. To filter the returned list by zone, specify ?zone={zone} in the URI.

This operation does not accept a request body.

### 4.26.1.2. Response

#### Example 4.215. List hosts: JSON response

```
{
  "hosts": [
    {
      "host_name": "787f4f6ddalb409bb8b2f9082349690e",
      "service": "compute",
      "zone": "nova"
    },
    {
      "host_name": "a98b433151084aee8bla986e28823b36",
      "service": "cert",
      "zone": "internal"
    },
    {
      "host_name": "c56158d13a884a87abf9171efb7de9d8",
      "service": "network",
      "zone": "internal"
    },
    {
      "host_name": "81d5cdcd0014918b3ebd3503a2e5c9a",
      "service": "scheduler",
      "zone": "internal"
    },
    {
      "host_name": "6e48bfela3304b7b86154326328750ae",
      "service": "volume",
      "zone": "internal"
    }
  ]
}
```

```
        "service": "conductor",
        "zone": "internal"
    },
    {
        "host_name": "39f55087a1024d1380755951c945ca69",
        "service": "cells",
        "zone": "internal"
    },
    {
        "host_name": "7a9a6cb4701f4dee9048fe0bc25d0ee5",
        "service": "consoleauth",
        "zone": "internal"
    }
]
```

### Example 4.216. List hosts: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hosts>
    <host zone="internal" host_name="02ff30050a0d4ce486e60e946a97d28f" service=
"conductor"/>
    <host zone="nova" host_name="69dd365b9eae459883a50d1fe7b3c40b" service=
"compute"/>
    <host zone="internal" host_name="059ca4da76cf49e8a3d08d2303542157" service=
"cert"/>
    <host zone="internal" host_name="ea70b9f0c530497fba8571deb4835ab0" service=
"network"/>
    <host zone="internal" host_name="7d52ad3988504fb18b0cdfd94dbd267b" service=
"scheduler"/>
    <host zone="internal" host_name="dc36bce962cc49c2ab916287bcf6c9d3" service=
"cells"/>
    <host zone="internal" host_name="4bd7bbcc80064d21a5fd0bd53318c6fa" service=
"consoleauth"/>
</hosts>
```

This operation does not return a response body.

## 4.26.2. Show host information

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-hosts/{host_name}	Shows information for a specified host.

**Normal response codes:** 200

### 4.26.2.1. Request

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

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

### 4.26.2.2. Response

#### Example 4.217. Show host information: JSON response

```
{
  "host": [
    {
      "resource": {
        "cpu": 1,
        "disk_gb": 1028,
        "host": "5ca60c6792a1442f9471ff575443f94d",
        "memory_mb": 8192,
        "project": "(total)"
      }
    },
    {
      "resource": {
        "cpu": 0,
        "disk_gb": 0,
        "host": "5ca60c6792a1442f9471ff575443f94d",
        "memory_mb": 512,
        "project": "(used_now)"
      }
    },
    {
      "resource": {
        "cpu": 0,
        "disk_gb": 0,
        "host": "5ca60c6792a1442f9471ff575443f94d",
        "memory_mb": 0,
        "project": "(used_max)"
      }
    }
  ]
}
```

### Example 4.218. Show host information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host>
  <resource>
    <project>(total)</project>
    <memory_mb>8192</memory_mb>
    <host>ecf3458ac6bf4a299cc2e0efa740f426</host>
    <cpu>1</cpu>
    <disk_gb>1028</disk_gb>
  </resource>
  <resource>
    <project>(used_now)</project>
    <memory_mb>512</memory_mb>
    <host>ecf3458ac6bf4a299cc2e0efa740f426</host>
    <cpu>0</cpu>
    <disk_gb>0</disk_gb>
  </resource>
  <resource>
    <project>(used_max)</project>
    <memory_mb>0</memory_mb>
    <host>ecf3458ac6bf4a299cc2e0efa740f426</host>
    <cpu>0</cpu>
    <disk_gb>0</disk_gb>
  </resource>
</host>
```

This operation does not return a response body.

## 4.26.3. Update host

Method	URI	Description
PUT	/v2/{tenant_id}/os-hosts/{host_name}	Enables a host or puts it in maintenance mode.

**Normal response codes:** 200

### 4.26.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{host_name}	String	The name of the host of interest to you.

#### Example 4.219. Update host: JSON request

```
{
    "status": "enable",
    "maintenance_mode": "disable"
}
```

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

Name	Type	Description
status	String <i>(Required)</i>	Valid values are enable or disable.
maintenance_mode	String <i>(Required)</i>	Valid values are enable or disable.

#### Example 4.220. Update host: XML request

```
<?xml version="1.0" encoding="UTF-8" ?>
<updates>
    <status>enable</status>
    <maintenance_mode>disable</maintenance_mode>
</updates>
```

This operation does not accept a request body.

### 4.26.3.2. Response

#### Example 4.221. Update host: JSON response

```
{
    "host": "0738dca90a8c43fdadd0be28715520e2",
    "maintenance_mode": "off_maintenance",
    "status": "enabled"
}
```

**Example 4.222. Update host: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<host status="enabled" maintenance_mode="off_maintenance" host=
"d85f05519b57457c83da18c39fa8e00d" />
```

This operation does not return a response body.

## 4.26.4. Start host

Method	URI	Description
GET	/v2/{tenant_id}/os-hosts/{host_name}/startup	Starts a host.

Normal response codes: 200

### 4.26.4.1. Request

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

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

### 4.26.4.2. Response

#### Example 4.223. Start host: JSON response

```
{
    "host": "57f5de2fa5b44f14974a4f50b9ffcbf8",
    "power_action": "startup"
}
```

#### Example 4.224. Start host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host host="7fae168ed18140d5a785ade2ac1bd420" power_action="startup"/>
```

This operation does not return a response body.

## 4.26.5. Shut down host

Method	URI	Description
GET	/v2/{tenant_id}/os-hosts/{host_name}/shutdown	Shuts down a host.

Normal response codes: 200

### 4.26.5.1. Request

This table shows the URI parameters for the shut down host request:

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

### 4.26.5.2. Response

#### Example 4.225. Shut down host: JSON response

```
{
    "host": "d2576862a2144ee6ad37d9e1938460a2",
    "power_action": "shutdown"
}
```

#### Example 4.226. Shut down host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host host="c82ca6da579440ac930ddee0e6530176" power_action="shutdown"/>
```

This operation does not return a response body.

## 4.26.6. Reboot host

Method	URI	Description
GET	/v2/{tenant_id}/os-hosts/{host_name}/reboot	Reboots a host.

Normal response codes: 200

### 4.26.6.1. Request

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

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

### 4.26.6.2. Response

#### Example 4.227. Reboot host: JSON response

```
{
  "host": "066bf157ab50481d8c607cfe584b2230",
  "power_action": "reboot"
}
```

#### Example 4.228. Reboot host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host host="ce8c5f8cde4a46ffb01dec0788ad4dfc" power_action="reboot" />
```

This operation does not return a response body.

## 4.27. Hypervisors (os-hypervisors)

Display extra statistical information from the machine that hosts the hypervisor through the API for the hypervisor (XenAPI or KVM/libvirt).

Method	URI	Description
GET	/v2/{tenant_id}/os-hypervisors	Lists hypervisors information for each server obtained through the hypervisor-specific API, such as libvirt or XenAPI.
GET	/v2/{tenant_id}/os-hypervisors/detail	Shows information for a specified hypervisor. Typically configured as an admin-only extension by using policy.json settings.
GET	/v2/{tenant_id}/os-hypervisors/statistics	Shows hypervisor statistics over all compute nodes.
GET	/v2/{tenant_id}/os-hypervisors/{hypervisor_hostname}	Shows the up time for a specified hypervisor.
GET	/v2/{tenant_id}/os-hypervisors/{hypervisor_hostname}/servers	Lists instances that belong to specific hypervisors.



## 4.27.1. List hypervisors

Method	URI	Description
GET	/v2/{tenant_id}/os-hypervisors	Lists hypervisors information for each server obtained through the hypervisor-specific API, such as libvirt or XenAPI.

Normal response codes: 200

### 4.27.1.1. Request

This table shows the URI parameters for the list hypervisors 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.

### 4.27.1.2. Response

#### Example 4.229. List hypervisors: JSON response

```
{
    "hypervisors": [
        {
            "hypervisor_hostname": "fake-mini",
            "id": 1
        }
    ]
}
```

This table shows the body parameters for the list hypervisors response:

Name	Type	Description
hypervisor_hostname	String <i>(Required)</i>	Hypervisor host name, retrieved from the per-server listing.
id	String <i>(Required)</i>	Hypervisor ID.

#### Example 4.230. List hypervisors: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisors>
    <hypervisor id="1" hypervisor_hostname="fake-mini"/>
</hypervisors>
```

This operation does not return a response body.

## 4.27.2. Show hypervisor information

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-hypervisors/detail	Shows information for a specified hypervisor. Typically configured as an admin-only extension by using policy.json settings.

**Normal response codes:** 200

### 4.27.2.1. Request

This table shows the URI parameters for the show hypervisor information 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.

### 4.27.2.2. Response

#### Example 4.231. Show hypervisor information: JSON response

```
{
  "hypervisors": [
    {
      "cpu_info": "?",
      "current_workload": 0,
      "disk_available_least": null,
      "free_disk_gb": 1028,
      "free_ram_mb": 7680,
      "hypervisor_hostname": "fake-mini",
      "hypervisor_type": "fake",
      "hypervisor_version": 1,
      "id": 1,
      "local_gb": 1028,
      "local_gb_used": 0,
      "memory_mb": 8192,
      "memory_mb_used": 512,
      "running_vms": 0,
      "service": {
        "host": "1e0d7892083548cfb347e782d3b20342",
        "id": 2
      },
      "vcpus": 1,
      "vcpus_used": 0
    }
  ]
}
```

#### Example 4.232. Show hypervisor information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisors>
  <hypervisor vcpus_used="0" hypervisor_type="fake" local_gb_used="0"
  hypervisor_hostname="fake-mini" memory_mb_used="512" memory_mb="8192"
```

```
current_workload="0" vcpus="1" cpu_info=? running_vms="0" free_disk_gb=
"1028" hypervisor_version="1" disk_available_least="None" local_gb="1028"
free_ram_mb="7680" id="1">
  <service host="4400f556a66d44ce95dfa61e75a23aaf" id="2"/>
</hypervisor>
</hypervisors>
```

This operation does not return a response body.

## 4.27.3. Show statistics for hypervisors

Method	URI	Description
GET	/v2/{tenant_id}/os-hypervisors/statistics	Shows hypervisor statistics over all compute nodes.

**Normal response codes:** 200

### 4.27.3.1. Request

This table shows the URI parameters for the show statistics for hypervisors 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.

### 4.27.3.2. Response

#### Example 4.233. Show statistics for hypervisors: JSON response

```
{
  "hypervisor_statistics": {
    "count": 1,
    "current_workload": 0,
    "disk_available_least": 0,
    "free_disk_gb": 1028,
    "free_ram_mb": 7680,
    "local_gb": 1028,
    "local_gb_used": 0,
    "memory_mb": 8192,
    "memory_mb_used": 512,
    "running_vms": 0,
    "vcpus": 1,
    "vcpus_used": 0
  }
}
```

#### Example 4.234. Show statistics for hypervisors: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisor_statistics count="1" vcpus_used="0" local_gb_used="0" memory_mb=
"8192" current_workload="0" vcpus="1" running_vms="0" free_disk_gb="1028"
disk_available_least="0" local_gb="1028" free_ram_mb="7680" memory_mb_used=
"512"/>
```

This operation does not return a response body.

## 4.27.4. Show hypervisor up time

Method	URI	Description
GET	/v2/{tenant_id}/os-hypervisors/{hypervisor_hostname}	Shows the up time for a specified hypervisor.

**Normal response codes:** 200

### 4.27.4.1. Request

This table shows the URI parameters for the show hypervisor up time request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{hypervisor_hostname}	String	The name of the host that runs the hypervisor.

This operation does not accept a request body.

### 4.27.4.2. Response

#### Example 4.235. Show hypervisor up time: JSON response

```
{
  "hypervisor": {
    "hypervisor_hostname": "fake-mini",
    "id": 1,
    "uptime": " 08:32:11 up 93 days, 18:25, 12 users,  load average: 0.20,
0.12, 0.14"
  }
}
```

#### Example 4.236. Show hypervisor up time: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisor uptime=" 08:32:11 up 93 days, 18:25, 12 users,  load average: 0.
20, 0.12, 0.14" id="1" hypervisor_hostname="fake-mini"/>
```

This operation does not return a response body.

## 4.27.5. List instances for hypervisors

Method	URI	Description
GET	/v2/{tenant_id}/os-hypervisors/{hypervisor_hostname}/servers	Lists instances that belong to specific hypervisors.

Normal response codes: 200

### 4.27.5.1. Request

This table shows the URI parameters for the list instances for hypervisors request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{hypervisor_hostname}	String	The name of the host that runs the hypervisor.

This operation does not accept a request body.

### 4.27.5.2. Response

#### Example 4.237. List instances for hypervisors: JSON response

```
{
  "hypervisors": [
    {
      "hypervisor_hostname": "fake-mini",
      "id": 1
    }
  ]
}
```

#### Example 4.238. List instances for hypervisors: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisors>
  <hypervisor id="1" hypervisor_hostname="fake-mini">
    <servers/>
  </hypervisor>
</hypervisors>
```

This operation does not return a response body.

## 4.28. Server actions (os-instance-actions)

List available actions for a specified server. Administrators can get details for a specified action for a specified server.

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-instance-actions	Lists available actions for a specified server. Deployers set permissions for this request in the <code>policy.json</code> file. By default, all users can list actions.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-instance-actions/{action_id}	Gets details for a specified action for a specified server instance. Deployers set permissions for this request in the policy.json file. By default, only administrators can get details for an action.

## 4.28.1. List server actions

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-instance-actions	Lists available actions for a specified server. Deployers set permissions for this request in the <code>policy.json</code> file. By default, all users can list actions.

**Normal response codes:** 200

### 4.28.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server instance for which you want to list actions.

This operation does not accept a request body.

### 4.28.1.2. Response

#### Example 4.239. List server actions: JSON response

```
{
  "instanceActions": [
    {
      "action": "resize",
      "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
      "message": "",
      "project_id": "842",
      "request_id": "req-25517360-b757-47d3-be45-0e8d2a01b36a",
      "start_time": "2012-12-05 01:00:00.000000",
      "user_id": "789"
    },
    {
      "action": "reboot",
      "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
      "message": "",
      "project_id": "147",
      "request_id": "req-3293a3f1-b44c-4609-b8d2-d81b105636b8",
      "start_time": "2012-12-05 00:00:00.000000",
      "user_id": "789"
    }
  ]
}
```

#### Example 4.240. List server actions: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<instanceActions>
  <instanceAction instance_uuid="b48316c5-71e8-45e4-9884-6c78055b9b13"
  user_id="789" start_time="2012-12-05 01:00:00.000000" request_id=
  "req-25517360-b757-47d3-be45-0e8d2a01b36a" action="resize" message=""
  project_id="842"/>
```

```
<instanceAction instance_uuid="b48316c5-71e8-45e4-9884-6c78055b9b13"
    user_id="789" start_time="2012-12-05 00:00:00.000000" request_id=
    "req-3293a3f1-b44c-4609-b8d2-d81b105636b8" action="reboot" message=""
    project_id="147"/>
</instanceActions>
```

This operation does not return a response body.

## 4.28.2. Get action details

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-instance-actions/{action_id}	Gets details for a specified action for a specified server instance. Deployers set permissions for this request in the policy.json file. By default, only administrators can get details for an action.

**Normal response codes:** 200, 203

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

### 4.28.2.1. Request

This table shows the URI parameters for the get action 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 of the server instance for which you want to list actions.
{action_id}	UUID	The UUID of the action for which you want to get details.

This operation does not accept a request body.

### 4.28.2.2. Response

#### Example 4.241. Get action details: JSON response

```
{
  "instanceAction": {
    "action": "reboot",
    "events": [
      {
        "event": "schedule",
        "finish_time": "2012-12-05 01:02:00.000000",
        "result": "Success",
        "start_time": "2012-12-05 01:00:02.000000",
        "traceback": ""
      },
      {
        "event": "compute_create",
        "finish_time": "2012-12-05 01:04:00.000000",
        "result": "Success",
        "start_time": "2012-12-05 01:03:00.000000",
        "traceback": ""
      }
    ],
    "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
    "message": "",
    "project_id": "147",
    "request_id": "req-3293a3f1-b44c-4609-b8d2-d81b105636b8",
    "start_time": "2012-12-05 00:00:00.000000",
    "user_id": "789"
  }
}
```

```
    }  
}
```

### Example 4.242. Get action details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<instanceAction instance_uuid="b48316c5-71e8-45e4-9884-6c78055b9b13" user_id=  
"789" start_time="2012-12-05T00:00:00.000000" request_id="req-3293a3f1-  
b44c-4609-b8d2-d81b105636b8" action="reboot" message="" project_id="147">  
  <events finish_time="2012-12-05 01:02:00.000000" start_time="2012-12-05  
  01:00:02.000000" traceback="" event="schedule" result="Success"/>  
  <events finish_time="2012-12-05 01:04:00.000000" start_time="2012-12-05  
  01:03:00.000000" traceback="" event="compute_create" result="Success"/>  
</instanceAction>
```

This operation does not return a response body.

## 4.29. Keypairs (os-keypairs)

Generate, import, and delete SSH keys.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-keypairs	Lists keypairs that are associated with the account.
<b>POST</b>	/v2/{tenant_id}/os-keypairs	Generates or imports a keypair.
<b>DELETE</b>	/v2/{tenant_id}/os-key- pairs/{keypair_name}	Deletes a keypair.
<b>GET</b>	/v2/{tenant_id}/os-key- pairs/{keypair_name}	Shows a keypair associated with the account.

## 4.29.1. List keypairs

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

**Normal response codes:** 200

### 4.29.1.1. Request

This table shows the URI parameters for the list keypairs 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.

### 4.29.1.2. Response

#### Example 4.243. List keypairs: JSON response

```
{
  "keypairs": [
    {
      "keypair": {
        "fingerprint": "15:b0:f8:b3:f9:48:63:71:cf:7b:5b:38:6d:44:2d:4a",
        "name": "keypair-601a2305-4f25-41ed-89c6-2a966fc8027a",
        "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAgQC+Eo/RZRNgaaGTkFs7I62ZjsI1O79Kk1KbMXi8F+KITD4bVQHHn+kV
+4gRgkgCRbd0DqoGfpDFs877DYX9n4z6FrAIZ4PES8TNKhatifpn9NdQYWA
+IkU8Cuv1EKGuFpKRI/k7JLos/gHi2hy7QUwgtRvcfvD/vgQZOvwmGR9Q== Generated by
Nova\n"
      }
    }
  ]
}
```

#### Example 4.244. List keypairs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<keypairs>
  <keypair>
    <public_key>ssh-rsa
      AAAAB3NzaC1yc2EAAAQABAAgQCWdUoGD7qz9kjLoY2L0S5CdhUS8RvQ1g620TgvmWE/
      bEKDLwaTIFEEpN/0huGk/nxvVZ6VOhv1eSKC3o9dz2NDk0C4sBsrvJ41uWd1hbq72sDGzVEkJ
      +925CraiosAbMpRK5Ea7UPWbR8laqrY1TsKtcuxiGJ936bOPIXWl2h6Q== Generated by Nova
    </public_key>
    <name>keypair-a4c7d228-218b-4c4c-9d99-62e7878ebb1b</name>
    <fingerprint>62:32:23:67:56:ee:6f:51:4c:03:ce:b8:00:f9:41:ff</fingerprint>
  </keypair>
</keypairs>
```

This operation does not return a response body.

## 4.29.2. Create or import keypair

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

**Normal response codes:** 200

### 4.29.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.245. Create or import keypair: XML request

```
<keypair>
    <name>keypair-96bbe50e-05e1-4d59-9115-4779a3ebcc2e</name>
    <public_key>ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAAgQDx8nkQv/zgGgB4rMYmIf
+6A416Rr+o/61HBQdW5aYd44bd8JttDCE/F/pNRr0lRE+PiqSPO8nDPHw0010JeMH9gYgnnFlyY3/
OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/0d3ac30bNH6Sw9JD9UZHcpSxsIbECHw==
    Generated by Nova</public_key>
</keypair>
```

#### Example 4.246. Create or import keypair: JSON request

```
{
    "keypair": {
        "name": "keypair-dab428fe-6186-4a14-b3de-92131f76cd39",
        "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAAgQDx8nkQv/
zgGgB4rMYmIf+6A416Rr+o/61HBQdW5aYd44bd8JttDCE/F/pNRr0lRE
+PiqSPO8nDPHw0010JeMH9gYgnnFlyY3/OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/
0d3ac30bNH6Sw9JD9UZHcpSxsIbECHw== Generated by Nova"
    }
}
```

This operation does not accept a request body.

### 4.29.2.2. Response

#### Example 4.247. Create or import keypair: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<keypair>
    <public_key>ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAAgQDx8nkQv/zgGgB4rMYmIf
+6A416Rr+o/61HBQdW5aYd44bd8JttDCE/F/pNRr0lRE+PiqSPO8nDPHw0010JeMH9gYgnnFlyY3/
OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/0d3ac30bNH6Sw9JD9UZHcpSxsIbECHw==
    Generated by Nova</public_key>
    <user_id>fake</user_id>
    <name>keypair-96bbe50e-05e1-4d59-9115-4779a3ebcc2e</name>
    <fingerprint>1e:2c:9b:56:79:4b:45:77:f9:ca:7a:98:2c:b0:d5:3c</fingerprint>
</keypair>
```

#### Example 4.248. 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-dab428fe-6186-4a14-b3de-92131f76cd39",
        "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAAgQDx8nkQv/zgGgB4rMYmIf+6A4l6Rr+o/61HBQdW5aYd44bd8JttDCE/F/pNRr01RE+PiqSP08nDPHw0010JeMH9gYgnnFlyY3/OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/0d3ac30bNH6Sw9JD9UZHcpSxsIbECHw== Generated by Nova",
        "user_id": "fake"
    }
}
```

This operation does not return a response body.

## 4.29.3. Delete keypair

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

**Normal response codes:** 202

### 4.29.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{keypair_name}	String	The name associated with a keypair.

This operation does not accept a request body.

## 4.29.4. Show keypair information

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

Normal response codes: 200

### 4.29.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{keypair_name}	String	The name associated with a keypair.

This operation does not accept a request body.

### 4.29.4.2. Response

#### Example 4.249. Show keypair information: JSON response

```
{
    "keypair": {
        "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAAgQDCSLxfzqB
+e5yHdUSXvbxBKkajjlfuhv
+GArseqPjfkKJ6no5echpin7dJp0FLXMJKxJZE3WWIRu25CQrJntmi7no27RkDfAGaTFbjz3DWY4A1HLeKAB5tFTwYQ
DY7UvKaawLh4Wvh5vMmXgF7AFzdkI28urwo+Q== nova@use03147k5-eth0\n",
        "name": "hpdefault",
        "fingerprint": "8b:2f:aa:b0:b8:97:dc:c8:50:aa:d4:8e:d0:34:61:d9"
    }
}
```

## 4.30. Migrations (os-migrations)

Administrators only. Fetch in-progress migrations for a region or a specified cell in a region.

Method	URI	Description
GET	/v2/{tenant_id}/os-migrations{?host,status,cell_name}	Enables an administrative user to fetch in-progress migrations for a region or specified cell in a region.

## 4.30.1. Get migrations

Method	URI	Description
GET	/v2/{tenant_id}/os-migrations{?host,status,cell_name}	Enables an administrative user to fetch in-progress migrations for a region or specified cell in a region.

**Normal response codes:** 200

### 4.30.1.1. Request

This table shows the URI parameters for the get migrations 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.

### 4.30.1.2. Response

#### Example 4.250. Get 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"
        }
    ]
}
```

This table shows the body parameters for the get migrations response:

Name	Type	Description
created_at	String <i>(Required)</i>	The created at time stamp.
dest_compute	String <i>(Required)</i>	The destination compute node.
dest_host	String <i>(Required)</i>	The destination host.
id	String <i>(Required)</i>	The migration ID.
instance_uuid	Uuid <i>(Required)</i>	The ID of the instance being migrated.
new_instance_type_id	String <i>(Required)</i>	The ID of the new instance type.
old_instance_type_id	String <i>(Required)</i>	The ID of the old instance type.
source_compute	String <i>(Required)</i>	The source compute node.
source_node	String <i>(Required)</i>	The source node.
status	String <i>(Required)</i>	The migration status.
updated_at	String <i>(Required)</i>	The updated at time stamp.

### Example 4.251. Get migrations: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<migrations>
  <migration dest_host="1.2.3.4" status="Done" old_instance_type_id="1"
    updated_at="2012-10-29 13:42:02" dest_compute="compute2" created_at=
    "2012-10-29 13:42:02" source_node="node1" instance_uuid="instance_id_123"
    dest_node="node2" id="1234" new_instance_type_id="2" source_compute=
    "compute1"/>
  <migration dest_host="5.6.7.8" status="Done" old_instance_type_id="5"
    updated_at="2013-10-22 13:42:02" dest_compute="compute20" created_at=
    "2013-10-22 13:42:02" source_node="node10" instance_uuid="instance_id_456"
    dest_node="node20" id="5678" new_instance_type_id="6" source_compute=
    "compute10"/>
</migrations>
```

This operation does not return a response body.

## 4.31. Networks (os-networks)

Show network information for or delete networks. Also, disassociate a network from a project if you use vlan mode.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}	Creates a network.
<b>GET</b>	/v2/{tenant_id}/os-networks	Lists networks that are available to the tenant.
<b>POST</b>	/v2/{tenant_id}/os-networks/add	Adds a specified network to a project.
<b>GET</b>	/v2/{tenant_id}/os-networks/{id}	Shows information for a specified network.
<b>DELETE</b>	/v2/{tenant_id}/os-networks/{id}	Deletes a specified network.
<b>POST</b>	/v2/{tenant_id}/os-networks/{id}/action	Associates a specified network with a host.
<b>POST</b>	/v2/{tenant_id}/os-networks/{id}/action	Disassociates the host from a specified network.
<b>POST</b>	/v2/{tenant_id}/os-networks/{id}/action	Disassociates a specified network from a project so that the network can be reused.
<b>POST</b>	/v2/{tenant_id}/os-networks/{id}/action	Disassociates the project from a specified network.

## 4.31.1. Create network

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

**Normal response codes:** 202

### 4.31.1.1. Request

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

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

### Example 4.252. Create network: JSON request

```
{
    "network": {
        "label": "new net 111",
        "cidr": "10.20.105.0/24"
    }
}
```

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

Name	Type	Description
network	Dict <i>(Required)</i>	A dictionary representation of a network.
label	String <i>(Required)</i>	Network label.
bridge	String <i>(Optional)</i>	VIFs on this network are connected to this bridge.
bridge_interface	String <i>(Optional)</i>	The bridge is connected to this interface.
cidr	String <i>(Optional)</i>	IPv4 subnet.
multi_host	String <i>(Optional)</i>	Multi host.
vlan	UUID <i>(Optional)</i>	Vlan id.
cidr_v6	String <i>(Optional)</i>	IPv6 subnet.
dns1	String <i>(Optional)</i>	First DNS.
dns2	String <i>(Optional)</i>	Second DNS.
gateway	String	IPv4 gateway.

Name	Type	Description
	(Optional)	
gateway_v6	String	IPv6 gateway.
	(Optional)	
project_id	String	Project id.
	(Optional)	

### Example 4.253. Create network: XML request

```
<network>
    <label>new net 111</label>
    <cidr>10.20.105.0/24</cidr>
</network>
```

This operation does not accept a request body.

### 4.31.1.2. Response

#### Example 4.254. 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_start": "10.20.105.2",
        "dns1": null,
        "dns2": null,
        "gateway": "10.20.105.1",
        "gateway_v6": null,
        "host": null,
        "id": "668687f9-d724-4976-a6f4-a6fd3ad83da3",
        "injected": null,
        "label": "new net 111",
        "multi_host": null,
        "netmask": "255.255.255.0",
        "netmask_v6": null,
        "priority": null,
        "project_id": null,
        "rxtx_base": null,
        "updated_at": null,
        "vlan": null,
        "vpn_private_address": null,
        "vpn_public_address": null,
        "vpn_public_port": null
    }
}
```

#### Example 4.255. Create network: XML response

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

```
<network>
  <bridge>None</bridge>
  <vpn_public_port>None</vpn_public_port>
  <dhcp_start>10.20.105.2</dhcp_start>
  <bridge_interface>None</bridge_interface>
  <updated_at>None</updated_at>
  <id>1bbbbed2b-0daa-47a1-b869-1981c29150b1</id>
  <cidr_v6>None</cidr_v6>
  <deleted_at>None</deleted_at>
  <gateway>10.20.105.1</gateway>
  <rxtx_base>None</rxtx_base>
  <label>new net 111</label>
  <priority>None</priority>
  <project_id>None</project_id>
  <vpn_private_address>None</vpn_private_address>
  <deleted>False</deleted>
  <vlan>None</vlan>
  <broadcast>10.20.105.255</broadcast>
  <netmask>255.255.255.0</netmask>
  <Injected>None</Injected>
  <cidr>10.20.105.0/24</cidr>
  <vpn_public_address>None</vpn_public_address>
  <multi_host>None</multi_host>
  <dns2>None</dns2>
  <created_at>None</created_at>
  <host>None</host>
  <gateway_v6>None</gateway_v6>
  <netmask_v6>None</netmask_v6>
  <dns1>None</dns1>
</network>
```

This operation does not return a response body.

## 4.31.2. List networks

Method	URI	Description
GET	/v2/{tenant_id}/os-networks	Lists networks that are available to the tenant.

**Normal response codes:** 200

### 4.31.2.1. Request

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

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

This operation does not accept a request body.

### 4.31.2.2. Response

#### Example 4.256. 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": "2011-08-15 06:19:19.387525",
      "deleted": false,
      "deleted_at": null,
      "dhcp_start": "10.0.0.3",
      "dns1": null,
      "dns2": null,
      "gateway": "10.0.0.1",
      "gateway_v6": null,
      "host": "nsokolov-desktop",
      "id": "20c8acc0-f747-4d71-a389-46d078ebf047",
      "injected": false,
      "label": "mynet_0",
      "multi_host": false,
      "netmask": "255.255.255.248",
      "netmask_v6": null,
      "priority": null,
      "project_id": "1234",
      "rxtx_base": null,
      "updated_at": "2011-08-16 09:26:13.048257",
      "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": "10.0.0.7",
      "cidr": "10.0.0.0/29",
      "cidr_v6": null,
      "created_at": "2011-08-15 06:19:19.387525",
      "deleted": false,
      "deleted_at": null,
      "dhcp_start": "10.0.0.3",
      "dns1": null,
      "dns2": null,
      "gateway": "10.0.0.1",
      "gateway_v6": null,
      "host": "nsokolov-desktop",
      "id": "20c8acc0-f747-4d71-a389-46d078ebf047",
      "injected": false,
      "label": "mynet_0",
      "multi_host": false,
      "netmask": "255.255.255.248",
      "netmask_v6": null,
      "priority": null,
      "project_id": "1234",
      "rxtx_base": null,
      "updated_at": "2011-08-16 09:26:13.048257",
      "vlan": 100,
      "vpn_private_address": "10.0.0.2",
      "vpn_public_address": "127.0.0.1",
      "vpn_public_port": 1000
    }
  ]
}
```

```
        "broadcast": "10.0.0.15",
        "cidr": "10.0.0.10/29",
        "cidr_v6": null,
        "created_at": "2011-08-15 06:19:19.885495",
        "deleted": false,
        "deleted_at": null,
        "dhcp_start": "10.0.0.11",
        "dns1": null,
        "dns2": null,
        "gateway": "10.0.0.9",
        "gateway_v6": null,
        "host": null,
        "id": "20c8acc0-f747-4d71-a389-46d078ebf000",
        "Injected": false,
        "label": "mynet_1",
        "multi_host": false,
        "netmask": "255.255.255.248",
        "netmask_v6": null,
        "priority": null,
        "project_id": null,
        "rxtx_base": null,
        "updated_at": null,
        "vlan": 101,
        "vpn_private_address": "10.0.0.10",
        "vpn_public_address": null,
        "vpn_public_port": 1001
    }
]
}
```

### Example 4.257. List networks: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<networks>
    <network>
        <brIDGE>br100</bridge>
        <vpn_public_port>1000</vpn_public_port>
        <dhcp_start>10.0.0.3</dhcp_start>
        <brIDGE_interface>eth0</bridge_interface>
        <updated_at>2011-08-16 09:26:13.048257</updated_at>
        <id>20c8acc0-f747-4d71-a389-46d078ebf047</id>
        <cidr_v6>None</cidr_v6>
        <deleted_at>None</deleted_at>
        <gateway>10.0.0.1</gateway>
        <rxtx_base>None</rxtx_base>
        <label>mynet_0</label>
        <priority>None</priority>
        <project_id>1234</project_id>
        <vpn_private_address>10.0.0.2</vpn_private_address>
        <deleted>False</deleted>
        <vlan>100</vlan>
        <broadcast>10.0.0.7</broadcast>
        <netmask>255.255.255.248</netmask>
        <Injected>False</Injected>
        <cidr>10.0.0.0/29</cidr>
        <vpn_public_address>127.0.0.1</vpn_public_address>
        <multi_host>False</multi_host>
        <dns2>None</dns2>
        <created_at>2011-08-15 06:19:19.387525</created_at>
        <host>nsokolov-desktop</host>
```

```
<gateway_v6>None</gateway_v6>
<netmask_v6>None</netmask_v6>
<dns1>None</dns1>
</network>
<network>
<bridge>br101</bridge>
<vpn_public_port>1001</vpn_public_port>
<dhcp_start>10.0.0.11</dhcp_start>
<bridge_interface>eth0</bridge_interface>
<updated_at>None</updated_at>
<id>20c8acc0-f747-4d71-a389-46d078ebf000</id>
<cidr_v6>None</cidr_v6>
<deleted_at>None</deleted_at>
<gateway>10.0.0.9</gateway>
<rxtx_base>None</rxtx_base>
<label>mynet_1</label>
<priority>None</priority>
<project_id>None</project_id>
<vpn_private_address>10.0.0.10</vpn_private_address>
<deleted>False</deleted>
<vlan>101</vlan>
<broadcast>10.0.0.15</broadcast>
<netmask>255.255.255.248</netmask>
<injected>False</injected>
<cidr>10.0.0.10/29</cidr>
<vpn_public_address>None</vpn_public_address>
<multi_host>False</multi_host>
<dns2>None</dns2>
<created_at>2011-08-15 06:19:19.885495</created_at>
<host>None</host>
<gateway_v6>None</gateway_v6>
<netmask_v6>None</netmask_v6>
<dns1>None</dns1>
</network>
</networks>
```

This operation does not return a response body.

## 4.31.3. Add network

Method	URI	Description
POST	/v2/{tenant_id}/os-networks/add	Adds a specified network to a project.

**Normal response codes:** 202

### 4.31.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	Uuid	The UUID of the network to add to the project. Specify None to choose a random available network.

#### Example 4.258. Add network: JSON request

```
{
    "id": "1"
}
```

#### Example 4.259. Add network: XML request

```
<id>1</id>
```

This operation does not accept a request body.

## 4.31.4. Show network information

Method	URI	Description
GET	/v2/{tenant_id}/os-networks/{id}	Shows information for a specified network.

**Normal response codes:** 200

### 4.31.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

This operation does not accept a request body.

### 4.31.4.2. Response

#### Example 4.260. Show network information: JSON response

```
{
  "network": {
    "bridge": "br100",
    "bridge_interface": "eth0",
    "broadcast": "10.0.0.7",
    "cidr": "10.0.0.0/29",
    "cidr_v6": null,
    "created_at": "2011-08-15 06:19:19.387525",
    "deleted": false,
    "deleted_at": null,
    "dhcp_start": "10.0.0.3",
    "dns1": null,
    "dns2": null,
    "gateway": "10.0.0.1",
    "gateway_v6": null,
    "host": "nsokolov-desktop",
    "id": "20c8acc0-f747-4d71-a389-46d078ebf047",
    "injected": false,
    "label": "mynet_0",
    "multi_host": false,
    "netmask": "255.255.255.248",
    "netmask_v6": null,
    "priority": null,
    "project_id": "1234",
    "rxtx_base": null,
    "updated_at": "2011-08-16 09:26:13.048257",
    "vlan": 100,
    "vpn_private_address": "10.0.0.2",
    "vpn_public_address": "127.0.0.1",
    "vpn_public_port": 1000
  }
}
```

**Example 4.261. Show network information: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<network>
    <bridge>br100</bridge>
    <vpn_public_port>1000</vpn_public_port>
    <dhcp_start>10.0.0.3</dhcp_start>
    <bridge_interface>eth0</bridge_interface>
    <updated_at>2011-08-16 09:26:13.048257</updated_at>
    <id>20c8acc0-f747-4d71-a389-46d078ebf047</id>
    <cidr_v6>None</cidr_v6>
    <deleted_at>None</deleted_at>
    <gateway>10.0.0.1</gateway>
    <rxtx_base>None</rxtx_base>
    <label>mynet_0</label>
    <priority>None</priority>
    <project_id>1234</project_id>
    <vpn_private_address>10.0.0.2</vpn_private_address>
    <deleted>False</deleted>
    <vlan>100</vlan>
    <broadcast>10.0.0.7</broadcast>
    <netmask>255.255.255.248</netmask>
    <injected>False</injected>
    <cidr>10.0.0.0/29</cidr>
    <vpn_public_address>127.0.0.1</vpn_public_address>
    <multi_host>False</multi_host>
    <dns2>None</dns2>
    <created_at>2011-08-15 06:19:19.387525</created_at>
    <host>nsokolov-desktop</host>
    <gateway_v6>None</gateway_v6>
    <netmask_v6>None</netmask_v6>
    <dns1>None</dns1>
</network>
```

This operation does not return a response body.

## 4.31.5. Delete network

Method	URI	Description
DELETE	/v2/{tenant_id}/os-networks/{id}	Deletes a specified network.

**Normal response codes:** 202

### 4.31.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

This operation does not accept a request body.

## 4.31.6. Associate host

Method	URI	Description
POST	/v2/{tenant_id}/os-networks/{id}/action	Associates a specified network with a host.

**Normal response codes:** 202

### 4.31.6.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

#### Example 4.262. Associate host: JSON request

```
{  
    "associate_host": "testHost"  
}
```

#### Example 4.263. Associate host: XML request

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

This operation does not accept a request body.

## 4.31.7. Disassociate host

Method	URI	Description
POST	/v2/{tenant_id}/os-networks/{id}/action	Disassociates the host from a specified network.

Normal response codes: 202

### 4.31.7.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

#### Example 4.264. Disassociate host: JSON request

```
{
    "disassociate_host": null
}
```

#### Example 4.265. Disassociate host: XML request

```
<disassociate_host/>
```

This operation does not accept a request body.

## 4.31.8. Disassociate network

Method	URI	Description
POST	/v2/{tenant_id}/os-networks/{id}/action	Disassociates a specified network from a project so that the network can be reused.

Normal response codes: 202

### 4.31.8.1. Request

This table shows the URI parameters for the disassociate network request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

#### Example 4.266. Disassociate network: JSON request

```
{
    "disassociate": null
}
```

#### Example 4.267. Disassociate network: XML request

```
<disassociate/>
```

This operation does not accept a request body.

## 4.31.9. Disassociate project

Method	URI	Description
POST	/v2/{tenant_id}/os-networks/{id}/action	Disassociates the project from a specified network.

Normal response codes: 202

### 4.31.9.1. Request

This table shows the URI parameters for the disassociate project request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

#### Example 4.268. Disassociate project: JSON request

```
{
    "disassociate_project": null
}
```

#### Example 4.269. Disassociate project: XML request

```
<disassociate_project/>
```

This operation does not accept a request body.

## 4.32. Quota sets (os-quota-sets)

Administrators only, depending on policy settings. View quotas for a tenant and view and update default quotas.

Method	URI	Description
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}	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 specified tenant and user.
POST	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a specified tenant/project 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 specified tenant and user.

## 4.32.1. Show quotas

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

Normal response codes: 200

### 4.32.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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.

### 4.32.1.2. Response

#### Example 4.270. 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 <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	The number of floating IP addresses allowed for each tenant.

Name	Type	Description
	(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 4.271. 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.

## 4.32.2. Update quotas

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

Normal response codes: 200

### 4.32.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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 4.272. 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 4.273. 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.

### 4.32.2.2. Response

#### Example 4.274. 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 4.275. 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.

## 4.32.3. Delete quotas

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Deletes quotas for a tenant so the quotas revert to default values.

**Normal response codes:** 200

### 4.32.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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.

## 4.32.4. Get default quotas

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

Normal response codes: 200

### 4.32.4.1. Request

This table shows the URI parameters for the get default quotas 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.

### 4.32.4.2. Response

#### Example 4.276. 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 4.277. 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.

## 4.32.5. 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 specified tenant and user.

Normal response codes: 200

### 4.32.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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 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.

### 4.32.5.2. Response

#### Example 4.278. 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 4.279. 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.

## 4.32.6. Update quotas for user

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

Normal response codes: 200

### 4.32.6.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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 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 4.280. 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 4.281. 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.

#### 4.32.6.2. Response

#### Example 4.282. 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 4.283. 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.

## 4.32.7. Delete quotas for user

Method	URI	Description
<b>DELETE</b>	/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

### 4.32.7.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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 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.

## 4.32.8. 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 specified tenant and user.

**Normal response codes:** 200

### 4.32.8.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project 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 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.

### 4.32.8.2. Response

#### Example 4.284. 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
    }
}
```

## 4.33. Rules for default security group (os-security-group-default-rules)

List, show information for, and create default security group rules.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-security-group-default-rules	Lists default security group rules.
<b>POST</b>	/v2/{tenant_id}/os-security-group-default-rules	Creates a default security group rule.
<b>GET</b>	/v2/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Shows information for a specified security group rule.
<b>DELETE</b>	/v2/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Deletes a specified security group rule.

## 4.33.1. List default security group rules

Method	URI	Description
GET	/v2/{tenant_id}/os-security-group-default-rules	Lists default security group rules.

**Normal response codes:** 200

### 4.33.1.1. 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.

### 4.33.1.2. Response

#### Example 4.285. 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
        }
    ]
}
```

This table shows the body parameters for the list default security group rules response:

Name	Type	Description
from_port	Int <i>(Required)</i>	Port at start of range.
id	UUID <i>(Required)</i>	Security group ID.
ip_protocol	String <i>(Required)</i>	The IP protocol: ICMP, TCP, or UDP.
ip_range	String <i>(Required)</i>	An IP range object.
cidr	String <i>(Required)</i>	The CIDR for address range.
to_port	Int	Port at end of range.

Name	Type	Description
	(Required)	

### Example 4.286. List default security group rules: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group_default_rules xmlns="http://docs.openstack.org/compute/api/v1.1">
    <security_group_default_rule id="1">
        <ip_protocol>TCP</ip_protocol>
        <from_port>80</from_port>
        <to_port>80</to_port>
        <ip_range>
            <cidr>10.10.10.0/24</cidr>
        </ip_range>
    </security_group_default_rule>
</security_group_default_rules>
```

This operation does not return a response body.

## 4.33.2. Create default security group rule

Method	URI	Description
POST	/v2/{tenant_id}/os-security-group-default-rules	Creates a default security group rule.

Normal response codes: 200

### 4.33.2.1. 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 4.287. Create default security group rule: JSON request

```
{
    "security_group_default_rule": {
        "ip_protocol": "TCP",
        "from_port": "80",
        "to_port": "80",
        "cidr": "10.10.12.0/24"
    }
}
```

This table shows the body parameters for the create default security group rule request:

Name	Type	Description
id	UUID (Required)	Security group name or ID.
ip_protocol	String (Required)	The IP protocol: ICMP, TCP, or UDP.
from_port	Int (Required)	Port at start of range.
to_port	Int (Required)	Port at end of range.
cidr	String (Required)	The CIDR for address range.

#### Example 4.288. Create default security group rule: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group_default_rule>
    <ip_protocol>TCP</ip_protocol>
    <from_port>80</from_port>
    <to_port>80</to_port>
    <cidr>10.10.12.0/24</cidr>
</security_group_default_rule>
```

This operation does not accept a request body.

### 4.33.2.2. Response

#### Example 4.289. 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
    }
}
```

This table shows the body parameters for the create default security group rule response:

Name	Type	Description
from_port	Int <i>(Required)</i>	Port at start of range.
id	UUID <i>(Required)</i>	Security group ID.
ip_protocol	String <i>(Required)</i>	The IP protocol: ICMP, TCP, or UDP.
ip_range	String <i>(Required)</i>	An IP range object.
cidr	String <i>(Required)</i>	The CIDR for address range.
to_port	Int <i>(Required)</i>	Port at end of range.

#### Example 4.290. Create default security group rule: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group_default_rule xmlns="http://docs.openstack.org/compute/api/v1.
1" id="1">
    <ip_protocol>TCP</ip_protocol>
    <from_port>80</from_port>
    <to_port>80</to_port>
    <ip_range>
        <cidr>10.10.10.0/24</cidr>
    </ip_range>
</security_group_default_rule>
```

This operation does not return a response body.

### 4.33.3. Show default security group rule information

Method	URI	Description
GET	/v2/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Shows information for a specified security group rule.

Normal response codes: 200

#### 4.33.3.1. Request

This table shows the URI parameters for the show default security group rule information 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.

#### 4.33.3.2. Response

##### Example 4.291. Show default security group rule: JSON response

```
{
    "security_group_default_rule": {
        "id": 1,
        "from_port": 80,
        "to_port": 80,
        "ip_protocol": "TCP",
        "ip_range": {
            "cidr": "10.10.10.0/24"
        }
    }
}
```

This table shows the body parameters for the show default security group rule information response:

Name	Type	Description
from_port	Int <i>(Required)</i>	Port at start of range.
id	UUID <i>(Required)</i>	Security group ID.
ip_protocol	String <i>(Required)</i>	The IP protocol: ICMP, TCP, or UDP.
ip_range	String <i>(Required)</i>	An IP range object.
cidr	String <i>(Required)</i>	The CIDR for address range.

Name	Type	Description
to_port	Int <i>(Required)</i>	Port at end of range.

### Example 4.292. Show default security group rule: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group_default_rule xmlns="http://docs.openstack.org/compute/api/v1.
1" id="1">
  <from_port>80</from_port>
  <to_port>80</to_port>
  <ip_protocol>TCP</ip_protocol>
  <ip_range>
    <cidr>10.10.10.0/24</cidr>
  </ip_range>
</security_group_default_rule>
```

This operation does not return a response body.

## 4.33.4. Delete default security group rule

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Deletes a specified security group rule.

Normal response codes: 204

### 4.33.4.1. 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.

## 4.34. Security groups (os-security-groups)

List, show information for, create, and delete security groups.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-security-groups	Lists security groups.
<b>POST</b>	/v2/{tenant_id}/os-security-groups	Creates a security group.
<b>GET</b>	/v2/{tenant_id}/os-security-groups/servers/{server_id}/os-security-groups	Lists security groups for a specified server.
<b>GET</b>	/v2/{tenant_id}/os-security-groups/{security_group_id}	Shows information for a specified security group.
<b>DELETE</b>	/v2/{tenant_id}/os-security-groups/{security_group_id}	Deletes a specified security group.

## 4.34.1. List security groups

Method	URI	Description
GET	/v2/{tenant_id}/os-security-groups	Lists security groups.

**Normal response codes:** 200

### 4.34.1.1. Request

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

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

This operation does not accept a request body.

### 4.34.1.2. Response

#### Example 4.293. List security groups: JSON response

```
{
  "security_groups": [
    {
      "description": "default",
      "id": 1,
      "name": "default",
      "rules": [],
      "tenant_id": "openstack"
    }
  ]
}
```

#### Example 4.294. List security group: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_groups xmlns="http://docs.openstack.org/compute/api/v1.1">
  <security_group tenant_id="openstack" id="1" name="default">
    <description>default</description>
    <rules/>
  </security_group>
</security_groups>
```

This operation does not return a response body.

## 4.34.2. Create security group

Method	URI	Description
POST	/v2/{tenant_id}/os-security-groups	Creates a security group.

**Normal response codes:** 200

### 4.34.2.1. Request

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

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

#### Example 4.295. Create security group: JSON request

```
{
    "addSecurityGroup": {
        "name": "test"
    }
}
```

#### Example 4.296. Create security group: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<addSecurityGroup>
    <name>test</name>
</addSecurityGroup>
```

This operation does not accept a request body.

### 4.34.2.2. Response

#### Example 4.297. Create security group: JSON response

```
{
    "security_group": {
        "description": "description",
        "id": 2,
        "name": "test",
        "rules": [],
        "tenant_id": "openstack"
    }
}
```

#### Example 4.298. Create security group: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group xmlns="http://docs.openstack.org/compute/api/v1.1" tenant_id=
"openstack" id="2" name="test">
    <description>
        description
    </description>
    <rules/>
```

```
</security_group>
```

This operation does not return a response body.

## 4.34.3. List security groups by server

Method	URI	Description
GET	/v2/{tenant_id}/os-security-groups/servers/{server_id}/os-security-groups	Lists security groups for a specified server.

Normal response codes: 200

### 4.34.3.1. Request

This table shows the URI parameters for the list security groups by server request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{server_id}	UUID	The UUID for the server of interest to you.

This operation does not accept a request body.

### 4.34.3.2. Response

#### Example 4.299. List security groups by server: JSON response

```
{
    "security_groups": [
        {
            "description": "default",
            "id": 1,
            "name": "default",
            "rules": [],
            "tenant_id": "openstack"
        }
    ]
}
```

#### Example 4.300. List security groups by server: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_groups xmlns="http://docs.openstack.org/compute/api/v1.1">
    <security_group tenant_id="openstack" id="1" name="default">
        <description>default</description>
        <rules/>
    </security_group>
</security_groups>
```

This operation does not return a response body.

## 4.34.4. Show security group information

Method	URI	Description
GET	/v2/{tenant_id}/os-security-groups/{security_group_id}	Shows information for a specified security group.

**Normal response codes:** 200

### 4.34.4.1. Request

This table shows the URI parameters for the show security group information request:

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

This operation does not accept a request body.

### 4.34.4.2. Response

#### Example 4.301. Show security group: JSON response

```
{
    "security_group": {
        "description": "default",
        "id": 1,
        "name": "default",
        "rules": [],
        "tenant_id": "openstack"
    }
}
```

#### Example 4.302. Show security group: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group xmlns="http://docs.openstack.org/compute/api/v1.1" tenant_id="openstack" id="1" name="default">
    <description>default</description>
    <rules/>
</security_group>
```

This operation does not return a response body.

## 4.34.5. Delete security group

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/os-security-groups/{security_group_id}	Deletes a specified security group.

**Normal response codes:** 202

### 4.34.5.1. Request

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

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

This operation does not accept a request body.

## 4.35. Server groups (os-server-groups)

List, show information for, create, and delete server groups.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-server-groups	Lists server groups.
<b>POST</b>	/v2/{tenant_id}/os-server-groups	Creates a server group.
<b>GET</b>	/v2/{tenant_id}/os-server-groups/{ServerGroup_id}	Shows details for a specified server group.
<b>DELETE</b>	/v2/{tenant_id}/os-server-groups/{ServerGroup_id}	Deletes a specified server group.

## 4.35.1. List server groups

Method	URI	Description
GET	/v2/{tenant_id}/os-server-groups	Lists server groups.

**Normal response codes:** 200200

### 4.35.1.1. Request

This table shows the URI parameters for the list server groups 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.

### 4.35.1.2. Response

#### Example 4.303. List server groups: JSON response

```
{
  "server_groups": [
    {
      "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
      "name": "test",
      "policies": [
        "anti-affinity"
      ],
      "members": [],
      "metadata": {}
    }
  ]
}
```

This table shows the body parameters for the list server groups response:

Name	Type	Description
server_groups	String <i>(Required)</i>	A server_groups object.
id	Uuid <i>(Required)</i>	The ID for the server group.
name	String <i>(Required)</i>	The name of the server group.
policies	Dict <i>(Required)</i>	A list of policies associated with the server group.
members	Dict <i>(Required)</i>	A list of members in the server group.
metadata	String <i>(Required)</i>	Associated metadata key-and-value pairs.

**Example 4.304. List server groups: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<server_groups xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
  <server_group id="5bbcc3c4-1da2-4437-a48a-66f15b1b13f9" name="test">
    <policies>
      <policy>anti-affinity</policy>
    </policies>
    <members/>
    <metadata/>
  </server_group>
</server_groups>
```

This operation does not return a response body.

## 4.35.2. Create server group

Method	URI	Description
POST	/v2/{tenant_id}/os-server-groups	Creates a server group.

**Normal response codes:** 200

### 4.35.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.305. Create server group: JSON request

```
{
    "server_group": {
        "name": "test",
        "policies": [
            "anti-affinity"
        ]
    }
}
```

This table shows the body parameters for the create server group request:

Name	Type	Description
name	String <i>(Required)</i>	The server group name. A non-empty string with no leading or trailing spaces. Maximum length is 255 characters.
policies	Dict <i>(Optional)</i>	A list of one or more policy names to associate with the server group. The list must contain at least one policy name.  Each policy name must be a non-empty string with no leading or trailing spaces. Maximum length is 255 characters.

#### Example 4.306. Create server group: XML request

```
<server_group name="test">
    <policies>
        <policy>anti-affinity</policy>
    </policies>
</server_group>
```

This operation does not accept a request body.

### 4.35.2.2. Response

#### Example 4.307. Create server group: JSON response

```
{
    "server_group": {
        "id": "5bcc3c4-1da2-4437-a48a-66f15b1b13f9",
```

```

        "name": "test",
        "policies": [
            "anti-affinity"
        ],
        "members": [],
        "metadata": {}
    }
}

```

This table shows the body parameters for the create server group response:

Name	Type	Description
server_groups	String <i>(Required)</i>	A server_groups object.
id	Uuid <i>(Required)</i>	The ID for the server group.
name	String <i>(Required)</i>	The name of the server group.
policies	Dict <i>(Required)</i>	A list of policies associated with the server group.
members	Dict <i>(Required)</i>	A list of members in the server group.
metadata	String <i>(Required)</i>	Associated metadata key-and-value pairs.

### Example 4.308. Create server group: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server_group xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1" id="5bbcc3c4-1da2-4437-a48a-66f15b1b13f9"
name="test">
    <policies>
        <policy>anti-affinity</policy>
    </policies>
    <members/>
    <metadata/>
</server_group>

```

This operation does not return a response body.

## 4.35.3. Show server group details

Method	URI	Description
GET	/v2/{tenant_id}/os-server-groups/{ServerGroup_id}	Shows details for a specified server group.

Normal response codes: 200200

### 4.35.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{ServerGroup_id}	String	The server group ID.

This operation does not accept a request body.

### 4.35.3.2. Response

#### Example 4.309. Show server group details: JSON response

```
{
  "server_group": {
    "id": "5bcc3c4-1da2-4437-a48a-66f15b1b13f9",
    "name": "test",
    "policies": [
      "anti-affinity"
    ],
    "members": [],
    "metadata": {}
  }
}
```

This table shows the body parameters for the show server group details response:

Name	Type	Description
server_groups	String (Required)	A server_groups object.
id	Uuid (Required)	The ID for the server group.
name	String (Required)	The name of the server group.
policies	Dict (Required)	A list of policies associated with the server group.
members	Dict (Required)	A list of members in the server group.
metadata	String (Required)	Associated metadata key-and-value pairs.

**Example 4.310. Show server group details: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<server_group xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1" id="5bbcc3c4-1da2-4437-a48a-66f15b1b13f9"
name="test">
<policies>
<policy>anti-affinity</policy>
</policies>
<members/>
<metadata/>
</server_group>
```

This operation does not return a response body.

## 4.35.4. Delete server group

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/os-server-groups/{ServerGroup_id}	Deletes a specified server group.

**Normal response codes:** 204

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

### 4.35.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{ServerGroup_id}	String	The server group ID.

This operation does not accept a request body.

## 4.36. Server password (os-server-password)

Get and reset the encrypted administrative password set through the metadata service.

Method	URI	Description
<b>GET</b>	/v2/servers/{server_id}/os-server-password	Gets the administrative password for a specified server.
<b>DELETE</b>	/v2/servers/{server_id}/os-server-password	Clears the encrypted copy of the password in the metadata server. This is done after the client has retrieved the password and knows it doesn't need it in the metadata server anymore. The password for the server remains the same.

## 4.36.1. Get server password

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

**Normal response codes:** 200

### 4.36.1.1. Request

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

Name	Type	Description
{server_id}	UUID	The UUID for the server of interest to you.

This operation does not accept a request body.

### 4.36.1.2. Response

#### Example 4.311. Get server password: JSON response

```
{
  "password": "xlozO3wLCBRWAa2yDjCCVx8vwNPypxnxpmRYDa/zErlQ+EzPe1S/
Gz6nfmC52m01OSCRUOmG7kqqgejPof6M7b0ezS387zjq4LSvvwp28zUknzy4YzfFGhnHAdai3TxUJ26pfQCYrq8UTzm
I1K2LsuipfxSJR7Wdke4zNXJjHHP2RfYsVbZ/k9ANu+Nz4iIH8/7Cacud/
pphH7EjrY6a4RZNrjQskrhKYed0YERpotyjYk1eDtRe72GrSiXteqCM4biaQ5w3ruS+AcX// 
PXk3uJ5kC7d67fPXaVz4WaQRYMg=="
}
```

#### Example 4.312. Get server password: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<password>xlozO3wLCBRWAa2yDjCCVx8vwNPypxnxpmRYDa/zErlQ+EzPe1S/
Gz6nfmC52m01OSCRUOmG7kqqgejPof6M7b0ezS387zjq4LSvvwp28zUknzy4YzfFGhnHAdai3TxUJ26pfQCYrq8UTzm
I1K2LsuipfxSJR7Wdke4zNXJjHHP2RfYsVbZ/k9ANu+Nz4iIH8/7Cacud/
pphH7EjrY6a4RZNrjQskrhKYed0YERpotyjYk1eDtRe72GrSiXteqCM4biaQ5w3ruS+AcX// 
PXk3uJ5kC7d67fPXaVz4WaQRYMg==</password>
```

This operation does not return a response body.

#### **4.36.2. Clear server password**

Method	URI	Description
<b>DELETE</b>	/v2/servers/{server_id}/os-server-password	Clears the encrypted copy of the password in the metadata server. This is done after the client has retrieved the password and knows it doesn't need it in the metadata server anymore. The password for the server remains the same.

## Normal response codes: 200

#### **4.36.2.1. Request**

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

Name	Type	Description
{server_id}	UUID	The UUID for the server of interest to you.

### Example 4.313. Clear server password: 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": "ICAgaICAgaDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcyBpbibQdXN0IHN1Y2ggYSBkaXJ1Y3Rpbd
= "
            }
        ]
    }
}
```

### Example 4.314. Clear server password: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1" imageRef="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b"
    flavorRef="http://openstack.example.com/openstack/flavors/1" name="new-server-test">
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <personality>
        <file path="/etc/banner.txt">
            ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
            dCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
            IGF0IHN1Y2ggYSBzcGV1ZC4uLkl0IGZlZWxzIGFuIGltcHVs
            c2lvbi4uLnRoaXMqaXMqdGhlIHBSYWNLIHrvIGdviG5vdv4q
        </file>
    </personality>
</server>
```

```

QnV0IHRoZSBza3kga25vd3MgdGh1IHJ1YXNvbnMgYW5kIHRo
ZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3VkcwygYW5kIH1v
dSB3aWxsIGtub3csIHRvbywgd2h1biB5b3UgbG1mdCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
b25zLiINCg0KLVJpY2hhcmQgQmFjaA==
</file>
</personality>
</server>

```

This operation does not accept a request body.

#### 4.36.2.2. Response

##### Example 4.315. Clear server password: JSON response

```
{
  "server": {
    "adminPass": "78AtBtuxTqZV",
    "id": "66fd64e1-de18-4506-bfb6-b5e73ef78a43",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/
66fd64e1-de18-4506-bfb6-b5e73ef78a43",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/
66fd64e1-de18-4506-bfb6-b5e73ef78a43",
        "rel": "bookmark"
      }
    ]
  }
}
```

##### Example 4.316. Clear server password: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1" id="b68e3354-0bla-4e92-a664-8b332cff27f5" adminPass=
"sLV7uLzmgoHu">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/
b68e3354-0bla-4e92-a664-8b332cff27f5" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/
b68e3354-0bla-4e92-a664-8b332cff27f5" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 4.37. Server shelve (servers)

Shelve running servers, restore shelved servers, and remove shelved servers.

Shelving is useful when you have an instance that you are not using but would like retain in your list of servers. For example, you can stop an instance at the end of a work week and resume work at the start of the next week. Use the `shelve` action to shelve a server. All associated data and resources are kept; however, anything still in memory is not retained.

You can restore a shelved instance by using the unshelve action. If a shelved instance is no longer needed, you can remove it by using the shelveOffload action.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Shelves a running server and changes its status to SHELVED_OFFLOADED.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Removes an instance from the server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Restores a shelved server.

## 4.37.1. Shelve server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Shelves a running server and changes its status to SHELVED_OFFLOADED.

All associated data and resources are kept; however, 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.

**Normal response codes:** 202

### 4.37.1.1. Request

This table shows the URI parameters for the shelve server 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 4.317. Shelve server: JSON request

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

This table shows the body parameters for the shelve server request:

Name	Type	Description
shelve	String <i>(Required)</i>	Specify the shelve action in the request body.

#### Example 4.318. Shelve server: XML request

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

This operation does not accept a request body.

## 4.37.2. Remove a shelved instance

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Removes an instance from the server.

Data and resource associations are deleted. If an instance is no longer needed, you can move that instance off the hypervisor to minimize resource usage.

**Normal response codes:** 202

### 4.37.2.1. Request

This table shows the URI parameters for the remove a shelved instance 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 4.319. Shelve server: JSON request

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

This table shows the body parameters for the remove a shelved instance request:

Name	Type	Description
shelveOffload	String <i>(Required)</i>	Specify the shelveOffload action in the request body.

#### Example 4.320. Shelve server: XML request

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

This operation does not accept a request body.

## 4.37.3. Restore shelved server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Restores a shelved server.

Normal response codes: 202

### 4.37.3.1. Request

This table shows the URI parameters for the restore shelved server 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 4.321. Unshelve server: JSON request

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

This table shows the body parameters for the restore shelved server request:

Name	Type	Description
unshelve	String (Required)	Specify the unshelve action in the request body.

#### Example 4.322. Unshelve server: XML request

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

This operation does not accept a request body.

## 4.38. Server start and stop (servers)

Start a stopped server or stop a running server.

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Starts a stopped server and changes its status to ACTIVE.
POST	/v2/{tenant_id}/servers/{server_id}/action	Stops a running server and changes its status to STOPPED.

## 4.38.1. Start server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Starts a stopped server and changes its status to ACTIVE.

Normal response codes: 202

### 4.38.1.1. Request

This table shows the URI parameters for the start server 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 4.323. Start server: JSON request

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

This table shows the body parameters for the start server request:

Name	Type	Description
os-start	String <i>(Required)</i>	Specify the os-start action in the request body.

#### Example 4.324. Start server: XML request

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

This operation does not accept a request body.

## 4.38.2. Stop server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Stops a running server and changes its status to STOPPED.

Normal response codes: 202

### 4.38.2.1. Request

This table shows the URI parameters for the stop server 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 4.325. Stop server: JSON request

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

This table shows the body parameters for the stop server request:

Name	Type	Description
os-stop	String <i>(Required)</i>	Specify the os-stop action in the request body.

#### Example 4.326. Stop server: XML request

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

This operation does not accept a request body.

## 4.39. Manage services (os-services)

List, enable, and disable Compute services in all hosts.

Method	URI	Description
GET	/v2/{tenant_id}/os-services	Lists running services.
PUT	/v2/{tenant_id}/os-services/enable	Enables scheduling for a service.
PUT	/v2/{tenant_id}/os-services/disable	Disables scheduling for a service.
PUT	/v2/{tenant_id}/os-services/disable-log-reason	Logs information to the service table about why a service was disabled.
GET	/v2/{tenant_id}/os-services/detail	Lists disabled services. If information exists, includes reasons why services were disabled.

## 4.39.1. List services

Method	URI	Description
GET	/v2/{tenant_id}/os-services	Lists running services.

**Normal response codes:** 200

### 4.39.1.1. Request

This table shows the URI parameters for the list services 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.

### 4.39.1.2. Response

#### Example 4.327. List services: JSON response

```
{
  "services": [
    {
      "binary": "nova-scheduler",
      "host": "host1",
      "state": "up",
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:02.000000",
      "zone": "internal"
    },
    {
      "binary": "nova-compute",
      "host": "host1",
      "state": "up",
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:05.000000",
      "zone": "nova"
    },
    {
      "binary": "nova-scheduler",
      "host": "host2",
      "state": "down",
      "status": "enabled",
      "updated_at": "2012-09-19T06:55:34.000000",
      "zone": "internal"
    },
    {
      "binary": "nova-compute",
      "host": "host2",
      "state": "down",
      "status": "disabled",
      "updated_at": "2012-09-18T08:03:38.000000",
      "zone": "nova"
    }
  ]
}
```

}

## 4.39.2. Enable scheduling for a service

Method	URI	Description
PUT	/v2/{tenant_id}/os-services/enable	Enables scheduling for a service.

**Normal response codes:** 200

### 4.39.2.1. Request

This table shows the URI parameters for the enable scheduling for a service request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.328. Enable scheduling for a service: JSON request

```
{  
    "host": "host1",  
    "binary": "nova-compute"  
}
```

### 4.39.2.2. Response

#### Example 4.329. Enable scheduling for a service: JSON response

```
{  
    "service": {  
        "host": "host1",  
        "binary": "nova-compute",  
        "status": "enabled"  
    }  
}
```

## 4.39.3. Disable scheduling for a service

Method	URI	Description
PUT	/v2/{tenant_id}/os-services/disable	Disables scheduling for a service.

Normal response codes: 200

### 4.39.3.1. Request

This table shows the URI parameters for the disable scheduling for a service request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.330. Disable scheduling for a service: JSON request

```
{
    "host": "host1",
    "binary": "nova-compute"
}
```

### 4.39.3.2. Response

#### Example 4.331. Disable scheduling for a service: JSON response

```
{
    "service": {
        "host": "host1",
        "binary": "nova-compute",
        "status": "disabled"
    }
}
```

## 4.39.4. Log disabled service information

Method	URI	Description
PUT	/v2/{tenant_id}/os-services/disable-log-reason	Logs information to the service table about why a service was disabled.

Normal response codes: 200

### 4.39.4.1. Request

This table shows the URI parameters for the log disabled service information request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.332. Log disabled service information: JSON request

```
{
  "host": "host1",
  "binary": "nova-compute",
  "disabled_reason": "test2"
}
```

This table shows the body parameters for the log disabled service information request:

Name	Type	Description
host	String <i>(Required)</i>	The host name.
binary	String <i>(Required)</i>	The service name.
disabled_reason	String <i>(Required)</i>	The reason the service was disabled.

#### Example 4.333. Log disabled service information: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<service host="host1" binary="nova-compute" disabled_reason="test2" />
```

This operation does not accept a request body.

### 4.39.4.2. Response

#### Example 4.334. Log disabled service information: JSON response

```
{
  "service": {
    "binary": "nova-compute",
    "host": "host1",
    "disabled_reason": "test2",
    "status": "disabled"
  }
}
```

{}

**Example 4.335. Log disabled service information: XML response**

```
<?xml version='1.0' encoding='UTF-8'?>
<service host="host1" binary="nova-compute" status="disabled" disabled_reason=
"test2" />
```

This operation does not return a response body.

## 4.39.5. List disabled services

Method	URI	Description
GET	/v2/{tenant_id}/os-services/detail	Lists disabled services. If information exists, includes reasons why services were disabled.

Normal response codes: 200

### 4.39.5.1. Request

This table shows the URI parameters for the list disabled services 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.

### 4.39.5.2. Response

#### Example 4.336. List disabled services: JSON response

```
{
  "services": [
    {
      "binary": "nova-scheduler",
      "host": "host1",
      "state": "up",
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:02.000000",
      "zone": "internal",
      "disabled_reason": "test1"
    },
    {
      "binary": "nova-compute",
      "host": "host1",
      "state": "up",
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:05.000000",
      "zone": "nova",
      "disabled_reason": "test2"
    },
    {
      "binary": "nova-scheduler",
      "host": "host2",
      "state": "down",
      "status": "enabled",
      "updated_at": "2012-09-19T06:55:34.000000",
      "zone": "internal",
      "disabled_reason": ""
    },
    {
      "binary": "nova-compute",
      "host": "host2",
      "state": "down",
      "status": "disabled",
      "updated_at": "2012-09-19T06:55:34.000000",
      "zone": "internal",
      "disabled_reason": ""
    }
  ]
}
```

```

        "updated_at": "2012-09-18T08:03:38.000000",
        "zone": "nova",
        "disabled_reason": "test4"
    }
]
}

```

### Example 4.337. List disabled services: XML response

```

<services>
    <service status="disabled" binary="nova-scheduler" zone="internal" state=
"up" host="host1" updated_at="2012-10-29T13:42:02.000000" disabled_reason=
"test1"/>
        <service status="disabled" binary="nova-compute" zone="nova" state="up"
host="host1" updated_at="2012-10-29T13:42:05.000000" disabled_reason="test2"/
>
    <service status="enabled" binary="nova-scheduler" zone="internal" state=
"down" host="host2" updated_at="2012-09-19T06:55:34.000000" disabled_reason=
"" />
        <service status="disabled" binary="nova-compute" zone="nova" state="down"
host="host2" updated_at="2012-09-18T08:03:38.000000" disabled_reason="test4"/
>
</services>

```

This operation does not return a response body.

## 4.40. Usage reports (os-simple-tenant-usage)

Report usage statistics on compute and storage resources.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/os-simple-ten-ant-usage	Lists usage information for all tenants.
<b>GET</b>	/v2/{tenant_id}/os-simple-ten-ant-usage/{tenant_id}	Gets usage information for a tenant.

## 4.40.1. List usage information for all tenants

Method	URI	Description
GET	/v2/{tenant_id}/os-simple-tenant-usage	Lists usage information for all tenants.

**Normal response codes:** 200

### 4.40.1.1. Request

This table shows the URI parameters for the list usage information for all tenants 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.

### 4.40.1.2. Response

#### Example 4.338. List usage information 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
    }
  ]
}
```

#### Example 4.339. List usage information for all tenants: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<tenant_usages>
  <tenant_usage>
    <tenant_id>openstack</tenant_id>
    <total_local_gb_usage>1.0</total_local_gb_usage>
    <total_vcpus_usage>1.0</total_vcpus_usage>
    <total_memory_mb_usage>512.0</total_memory_mb_usage>
    <total_hours>1.0</total_hours>
    <start>2012-10-08 21:10:51.902640</start>
    <stop>2012-10-08 22:10:51.902640</stop>
    <server_usages/>
  </tenant_usage>
</tenant_usages>
```

This operation does not return a response body.

## 4.40.2. Get tenant usage information

Method	URI	Description
GET	/v2/{tenant_id}/os-simple-tenant-usage/{tenant_id}	Gets usage information for a tenant.

**Normal response codes:** 200

### 4.40.2.1. Request

This table shows the URI parameters for the get tenant usage information request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{tenant_id}	String	The tenant ID.

This operation does not accept a request body.

### 4.40.2.2. Response

#### Example 4.340. Get tenant usage information: JSON response

```
{
  "tenant_usage": {
    "server_usages": [
      {
        "ended_at": null,
        "flavor": "ml.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
  }
}
```

#### Example 4.341. Get tenant usage information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<tenant_usage>
  <tenant_id>openstack</tenant_id>
```

```
<total_local_gb_usage>1.0</total_local_gb_usage>
<total_vcpus_usage>1.0</total_vcpus_usage>
<total_memory_mb_usage>512.0</total_memory_mb_usage>
<total_hours>1.0</total_hours>
<start>2012-10-08 20:10:51.902640</start>
<stop>2012-10-08 21:10:51.902640</stop>
<server_usages>
  <server_usage>
    <instance_id>e4521f3b-d9ad-4454-be8a-e2732f0630ef</instance_id>
    <name>new-server-test</name>
    <hours>1.0</hours>
    <memory_mb>512</memory_mb>
    <local_gb>1</local_gb>
    <vcpus>1</vcpus>
    <tenant_id>openstack</tenant_id>
    <flavor>m1.tiny</flavor>
    <started_at>2012-10-08 20:10:51.854331</started_at>
    <ended_at>None</ended_at>
    <state>active</state>
    <uptime>3600</uptime>
  </server_usage>
</server_usages>
</tenant_usage>
```

This operation does not return a response body.

## 4.41. Virtual interfaces (os-virtual-interfaces)

List the virtual interfaces for a specified server instance.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Lists the virtual interfaces for a specified instance.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Shows the virtual interface for a specified instance. Includes the OS-EXT-VIF-NET:net_id attribute that shows to which network the interface is attached.

## 4.41.1. List virtual interfaces

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Lists the virtual interfaces for a specified instance.

**Normal response codes:** 202

### 4.41.1.1. Request

This table shows the URI parameters for the list virtual 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.

### 4.41.1.2. Response

#### Example 4.342. List virtual interfaces: JSON response

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

#### Example 4.343. List virtual interfaces: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<virtual_interfaces xmlns="http://docs.openstack.org/compute/api/v1.1">
    <virtual_interface id="94edf7aa-565a-469a-8f45-656b4acf8229" mac_address=
    "fa:16:3e:7d:31:9a"/>
</virtual_interfaces>
```

This operation does not return a response body.

## 4.41.2. Show virtual interface and attached network

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Shows the virtual interface for a specified instance. Includes the OS-EXT-VIF-NET:net_id attribute that shows to which network the interface is attached.

Normal response codes: 202

### 4.41.2.1. Request

This table shows the URI parameters for the show virtual interface and attached network 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.

### 4.41.2.2. Response

#### Example 4.344. Show virtual interface and attached network: JSON response

```
{
    "virtual_interfaces": [
        {
            "id": "cec8b9bb-5d22-4104-b3c8-4c35db3210a6",
            "mac_address": "fa:16:3e:3c:ce:6f",
            "OS-EXT-VIF-NET:net_id": "cec8b9bb-5d22-4104-b3c8-4c35db3210a7"
        }
    ]
}
```

#### Example 4.345. Show virtual interface and attached network: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<virtual_interfaces
    xmlns:OS-EXT-VIF-NET="http://docs.openstack.org/compute/ext/extended-
    virtual-interfaces-net/api/v1.1"
    xmlns="http://docs.openstack.org/compute/api/v1.1">
    <virtual_interface id="94edf7aa-565a-469a-8f45-656b4acf8229"
        mac_address="fa:16:3e:7d:31:9a"
        OS-EXT-VIF-NET:net_id="94edf7aa-565a-469a-8f45-656b4acf8230"/>
</virtual_interfaces>
```

This operation does not return a response body.

## 4.42. Volume extension (os-volumes, os-snapshots)

Manage volumes and snapshots for use with the Compute API.

Method	URI	Description
<b>GET</b>	/v1.1/{tenant_id}/os-volumes	Lists the volumes associated with the account.
<b>GET</b>	/v1.1/{tenant_id}/os-volumes/detail	Lists details for a specified volume.
<b>POST</b>	/v1.1/{tenant_id}/os-volumes/{volume_id}	Creates a volume.
<b>GET</b>	/v1.1/{tenant_id}/os-volumes/{volume_id}	Shows information for a specified volume.
<b>DELETE</b>	/v1.1/{tenant_id}/os-volumes/{volume_id}	Deletes a specified volume.
<b>GET</b>	/v1.1/{tenant_id}/os-volume-types	Lists volume types.
<b>GET</b>	/v1.1/{tenant_id}/os-volume-types/{volume_type_id}	Shows information for a specified volume type.
<b>POST</b>	/v1.1/{tenant_id}/os-snapshots	Creates a snapshot.
<b>GET</b>	/v1.1/{tenant_id}/os-snapshots	Lists snapshots.
<b>GET</b>	/v1.1/{tenant_id}/os-snapshots/detail	Lists details for a specified snapshot.
<b>GET</b>	/v1.1/{tenant_id}/os-snapshots/{snapshot_id}	Shows information for a specified snapshot.
<b>DELETE</b>	/v1.1/{tenant_id}/os-snapshots/{snapshot_id}	Deletes a specified snapshot from the account.

## 4.42.1. List volumes

Method	URI	Description
GET	/v1.1/{tenant_id}/os-volumes	Lists the volumes associated with the account.

**Normal response codes:** 200

### 4.42.1.1. 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.

### 4.42.1.2. Response

#### Example 4.346. 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"
    }
  ]
}
```

#### Example 4.347. 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>
```

This operation does not return a response body.

## 4.42.2. List details for volumes

Method	URI	Description
GET	/v1.1/{tenant_id}/os-volumes/detail	Lists details for a specified volume.

**Normal response codes:** 200

### 4.42.2.1. Request

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

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

This operation does not accept a request body.

### 4.42.2.2. Response

#### Example 4.348. List details for 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"
    }
  ]
}
```

#### Example 4.349. List details for 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>
```

This operation does not return a response body.

## 4.42.3. Create volume

Method	URI	Description
POST	/v1.1/{tenant_id}/os-volumes/{volume_id}	Creates a volume.

**Normal response codes:** 201

### 4.42.3.1. 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.
{volume_id}	String	The unique identifier for a volume.

#### Example 4.350. 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 table shows the body parameters for the create volume request:

Name	Type	Description
volume	String <i>(Required)</i>	A partial representation of a volume that is used to create a volume.

#### Example 4.351. 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>
```

This operation does not accept a request body.

### 4.42.3.2. Response

#### Example 4.352. Create volume: JSON response

```
{  
    "volume": {  
        "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"  
    }  
}
```

#### Example 4.353. 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"  
        created_at="2012-02-14T20:53:07Z">  
    <metadata>  
        <meta key="contents">junk</meta>  
    </metadata>  
</volume>
```

This operation does not return a response body.

## 4.42.4. Show volume information

Method	URI	Description
GET	/v1.1/{tenant_id}/os-volumes/{volume_id}	Shows information for a specified volume.

**Normal response codes:** 200

### 4.42.4.1. Request

This table shows the URI parameters for the show volume information 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.

### 4.42.4.2. Response

#### Example 4.354. Show volume information: JSON response

```
{
    "volume": {
        "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"
    }
}
```

#### Example 4.355. Show volume information: 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"
        created_at="2012-02-14T20:53:07Z">
    <metadata>
        <meta key="contents">junk</meta>
    </metadata>
</volume>
```

This operation does not return a response body.

## 4.42.5. Delete volume

Method	URI	Description
DELETE	/v1.1/{tenant_id}/os-volumes/{volume_id}	Deletes a specified volume.

Normal response codes: 202

### 4.42.5.1. 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.

### 4.42.5.2. Response

#### Example 4.356. Delete volume: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Date: Fri, 02 Dec 2011 00:39:32 GMT
```

This operation does not return a response body.

## 4.42.6. List volume types

Method	URI	Description
GET	/v1.1/{tenant_id}/os-volume-types	Lists volume types.

**Normal response codes:** 200

### 4.42.6.1. 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.

### 4.42.6.2. Response

#### Example 4.357. 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": {}
        }
    ]
}
```

#### Example 4.358. 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>
```

This operation does not return a response body.

## 4.42.7. Show volume type

Method	URI	Description
GET	/v1.1/{tenant_id}/os-volume-types/{volume_type_id}	Shows information for a specified volume type.

**Normal response codes:** 200

### 4.42.7.1. Request

This table shows the URI parameters for the show volume type 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.

### 4.42.7.2. Response

#### Example 4.359. Show volume type: JSON response

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

#### Example 4.360. Show 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>
```

This operation does not return a response body.

## 4.42.8. Create snapshot

Method	URI	Description
POST	/v1.1/{tenant_id}/os-snapshots	Creates a snapshot.

**Normal response codes:** 201

### 4.42.8.1. 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 4.361. Create snapshot: JSON request

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

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

Name	Type	Description
snapshot	String <i>(Required)</i>	A partial representation of a snapshot that is used to create a snapshot.

#### Example 4.362. 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" />
```

This operation does not accept a request body.

### 4.42.8.2. Response

#### Example 4.363. 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"
    }
}
```

#### Example 4.364. 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" />
```

This operation does not return a response body.

## 4.42.9. List snapshots

Method	URI	Description
GET	/v1.1/{tenant_id}/os-snapshots	Lists snapshots.

**Normal response codes:** 200

### 4.42.9.1. 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.

### 4.42.9.2. Response

#### Example 4.365. 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"
    },
    {
      "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"
    }
  ]
}
```

#### Example 4.366. 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" />
  <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>
```

This operation does not return a response body.

## 4.42.10. List details for snapshots

Method	URI	Description
GET	/v1.1/{tenant_id}/os-snapshots/detail	Lists details for a specified snapshot.

**Normal response codes:** 200

### 4.42.10.1. Request

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

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

This operation does not accept a request body.

### 4.42.10.2. Response

#### Example 4.367. List details for 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"
    },
    {
      "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"
    }
  ]
}
```

#### Example 4.368. List details for 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" />
```

```
<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>
```

This operation does not return a response body.

## 4.42.11. Show snapshot

Method	URI	Description
GET	/v1.1/{tenant_id}/os-snapshots/{snapshot_id}	Shows information for a specified snapshot.

**Normal response codes:** 200

### 4.42.11.1. Request

This table shows the URI parameters for the show 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.

### 4.42.11.2. Response

#### Example 4.369. Show 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"
  }
}
```

#### Example 4.370. Show 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" />
```

This operation does not return a response body.

## 4.42.12. Delete snapshot

Method	URI	Description
<b>DELETE</b>	/v1.1/{tenant_id}/os-snapshots/{snapshot_id}	Deletes a specified snapshot from the account.

This operation is asynchronous. You must list snapshots repeatedly to determine whether the snapshot was deleted.

**Normal response codes:** 202

### 4.42.12.1. 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.

### 4.42.12.2. Response

#### Example 4.371. Delete snapshot: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Date: Mon, 05 Dec 2011 16:23:10 GMT
```

This operation does not return a response body.

## 4.43. Volume attachments (os-volume\_attachments)

Attach volumes created through the volume API to server instances. Also, list volume attachments for a server instance, get volume details for a volume attachment, and delete a volume attachment.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Attaches a volume to the specified server.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Lists the volume attachments for a specified server.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Shows details for the specified volume attachment.
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Deletes the specified volume attachment from a specified server.

## 4.43.1. Attach volume

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Attaches a volume to the specified server.

**Normal response codes:** 202

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

### 4.43.1.1. Request

This table shows the URI parameters for the attach volume 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.
{volumeId}	String	ID of the volume to attach.
{device}	String	Name of the device such as, /dev/vdb. Use "auto" for auto-assign (if supported).
{volumeAttachment}	String	A dictionary representation of a volume attachment.

#### Example 4.372. Attach volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volumeAttachment volumeId="a26887c6-c47b-4654-abb5-dfadf7d3f803" device="/dev/vdd" />
```

#### Example 4.373. Attach volume: JSON request

```
{
    "volumeAttachment": {
        "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
        "device": "/dev/vdd"
    }
}
```

### 4.43.1.2. Response

#### Example 4.374. Attach volume: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumeAttachment device="/dev/vdd" serverId="20f0cb44-7b00-4019-a612-364777cd2931" id="a26887c6-c47b-4654-abb5-dfadf7d3f803" volumeId="a26887c6-c47b-4654-abb5-dfadf7d3f803"/>
```

#### Example 4.375. Attach volume: JSON response

```
{
    "volumeAttachment": {
        "device": "/dev/vdd",
```

```
        "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
        "serverId": "0c92f3f6-c253-4c9b-bd43-e880a8d2eb0a",
        "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
    }
}
```

## 4.43.2. List volume attachments

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Lists the volume attachments for a specified server.

**Normal response codes:** 202

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

### 4.43.2.1. Request

This table shows the URI parameters for the list volume attachments 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.

### 4.43.2.2. Response

#### Example 4.376. List volume attachments: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumeAttachments>
  <volumeAttachment device="/dev/sdd" serverId=
    "4335bab6-6086-4247-8274-8b8b048edaaa" id="a26887c6-c47b-4654-abb5-
    dfadf7d3f803" volumeId="a26887c6-c47b-4654-abb5-dfadf7d3f803"/>
  <volumeAttachment device="/dev/sdc" serverId=
    "4335bab6-6086-4247-8274-8b8b048edaaa" id="a26887c6-c47b-4654-abb5-
    dfadf7d3f804" volumeId="a26887c6-c47b-4654-abb5-dfadf7d3f804"/>
</volumeAttachments>
```

#### Example 4.377. List volume attachments: JSON response

```
{
  "volumeAttachments": [
    {
      "device": "/dev/sdd",
      "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
      "serverId": "4d8c3732-a248-40ed-bebc-539a6ffd25c0",
      "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
    },
    {
      "device": "/dev/sdc",
      "id": "a26887c6-c47b-4654-abb5-dfadf7d3f804",
      "serverId": "4d8c3732-a248-40ed-bebc-539a6ffd25c0",
      "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f804"
    }
  ]
}
```



### 4.43.3. Show volume attachment details

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Shows details for the specified volume attachment.

**Normal response codes:** 202

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

#### 4.43.3.1. Request

This table shows the URI parameters for the show volume attachment 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}	String	Volume attachment ID.

This operation does not accept a request body.

#### 4.43.3.2. Response

##### Example 4.378. Show volume attachment details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumeAttachment device="/dev/sdd" serverId="20f12907-3993-44f7-a680-d51e2ceedbd9" id="a26887c6-c47b-4654-abb5-dfadf7d3f803" volumeId="a26887c6-c47b-4654-abb5-dfadf7d3f803" />
```

##### Example 4.379. Show volume attachment details: JSON response

```
{
  "volumeAttachment": {
    "device": "/dev/sdd",
    "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
    "serverId": "2390fb4d-1693-45d7-b309-e29c4af16538",
    "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
  }
}
```

## 4.43.4. Delete volume attachment

Method	URI	Description
<b>DELETE</b>	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Deletes the specified volume attachment from a specified server.

**Normal response codes:** 202

**Error response codes:** computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), buildInProgress (409)

### 4.43.4.1. Request

This table shows the URI parameters for the delete volume attachment 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}	String	Volume attachment ID.

This operation does not accept a request body.

## 4.44. Servers with block device mapping format (servers)

Create a server with a block device mapping.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/servers{?changes-since,image,flavor,name,marker,limit,status,host}	Lists IDs, names, and links for all servers.
<b>POST</b>	/v2/{tenant_id}/servers	Creates a server with a block device mapping.

## 4.44.1. List servers

Method	URI	Description
GET	/v2/{tenant_id}/servers{?changes-since,image,flavor,name,marker,limit,status,host}	Lists IDs, names, and links for all servers.

**Normal response codes:** 200, 203

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

### 4.44.1.1. Request

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

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

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

Name	Type	Description
changes-since	DateTime <i>(Optional)</i>	A time/date stamp for when the server last changed status.
image	AnyURI <i>(Optional)</i>	Name of the image in URL format.
flavor	AnyURI <i>(Optional)</i>	Name of the flavor in URL format.
name	String <i>(Optional)</i>	Name of the server as a string.
marker	UUID <i>(Optional)</i>	UUID of the server at which you want to set a marker.
limit	Int <i>(Optional)</i>	Integer value for the limit of values to return.
status	ServerStatus <i>(Optional)</i>	Value of the status of the server so that you can filter on "ACTIVE" for example.
host	String <i>(Optional)</i>	Name of the host as a string.

This operation does not accept a request body.

### 4.44.1.2. Response

#### Example 4.380. List servers: JSON response

```
{
```

```

"servers": [
    {
        "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/
servers/616fb98f-46ca-475e-917e-2563e5a8cd19",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
616fb98f-46ca-475e-917e-2563e5a8cd19",
                "rel": "bookmark"
            }
        ],
        "name": "new-server-test"
    }
]
}

```

This table shows the body parameters for the list servers response:

Name	Type	Description
servers	Servers WithOnly IDsNames Links  <i>(Required)</i>	List of servers.
next	AnyURI  <i>(Optional)</i>	Moves to the next item in the list.
previous	AnyURI  <i>(Optional)</i>	Moves to the previous item in the list.

### Example 4.381. List servers: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
    <server name="new-server-test" id="b626796d-d585-4874-b178-78c65289bba4">
        <atom:link href="http://openstack.example.com/v2/openstack/servers/
b626796d-d585-4874-b178-78c65289bba4" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/servers/b626796d-
d585-4874-b178-78c65289bba4" rel="bookmark"/>
    </server>
</servers>

```

## 4.44.2. Create server

Method	URI	Description
POST	/v2/{tenant_id}/servers	Creates a server with a block device mapping.

**Normal response codes:** 202

**Error response codes:** computeFault (400, 500, ...), UnprocessableEntity (422), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), NetworkNotFound (400), serverCapacityUnavailable (503)

### 4.44.2.1. Request

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

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

### Example 4.382. Create server: 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": "ICAgICAgDQoIQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3Zlc
yBpbBqdxN0IHN1Y2ggYSBkaXJ1Y3RpB2=",
                "encoding": "base64"
            }
        ],
        "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",
                "destination_type": "volume",
                "uuid": "fake-volume-id-1",
                "boot_index": "0"
            }
        ]
    }
}
```

```

    }
}
```

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

Name	Type	Description
security_groups  <i>(Optional)</i>	String	A security_groups object.
security_group  <i>(Optional)</i>	String	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.
user_data  <i>(Optional)</i>	String	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone  <i>(Optional)</i>	String	The availability zone in which to launch the server.
server  <i>(Required)</i>	ServerFor Create	server.
imageRef  <i>(Required)</i>	String	The image reference for the desired image for your server instance. Specify as an ID or full URL.
flavorRef  <i>(Required)</i>	String	The flavor reference for the desired flavor for your server instance. Specify as an ID or full URL.
networks  <i>(Optional)</i>	String	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant.  Optionally, you can create one or more NICs on the server.  To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a networks object.  To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a networks object.  You can specify multiple NICs on the server.
uuid  <i>(Optional)</i>	String	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a networks object. Required if you omit the <code>port</code> attribute.
port  <i>(Optional)</i>	String	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a networks object. Required if you omit the <code>uuid</code> attribute.
fixed_ip  <i>(Optional)</i>	String	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
name  <i>(Required)</i>	String	The server name.
metadata  <i>(Optional)</i>	String	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
personality  <i>(Optional)</i>	String	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
config_drive  <i>(Optional)</i>	String	Enables metadata injection in a server through a configuration drive.

Name	Type	Description
		To enable a configuration drive, specify true. Otherwise, specify false.
block_device_mapping_v2	BlockDeviceMapping <i>(Required)</i>	block_device_mapping_v2.
device_name	String <i>(Required)</i>	A device name where the volume is attached in the system at /dev/dev_name. This value is typically vda.
source_type	String <i>(Required)</i>	Either snap or any other value, including a blank string. snap means that the volume was created from a snapshot.
destination_type	String <i>(Required)</i>	Either snap or any other value, including a blank string. snap means that the volume was created from a snapshot.
delete_on_termination	Bool <i>(Required)</i>	Set to True (or 1) to delete the volume when the instance is deleted. Set to False (or 0) to retain the volume when the instance is deleted.
guest_format	String <i>(Required)</i>	Guest format.
uuid	Uuid <i>(Required)</i>	The device ID.
boot_index	String <i>(Required)</i>	The boot index.

### Example 4.383. Create server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1" imageRef="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" flavorRef="http://openstack.example.com/openstack/flavors/1" name="new-server-test">
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <personality>
        <file path="/etc/banner.txt">
            ICAgICAgDQoiQSBjbg91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
            dCBtb3ZlcYBpb1BqdXN0IHN1Y2ggYSBkaXJ1Y3RpB24gYW5k
            IGF0IHN1Y2ggYSBzcgV1ZC4uLk10IGZlZWxzIGFuIGltcHVs
            c2lvbi4uLnRoaXMgaXMgdGh1IHByWn1IHRvIGdvIG5vdy4g
            QnV0IHRoZSBza3kga25vd3MgdGh1IHJ1YXNvbMgYW5kIHRo
            ZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3VkcwggYW5kIHLv
            dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbG1mdCB5b3Vy
            c2VsZiBoawdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
            b25zLiINCg0KLVJpY2hhcmQgQmFjaA==
        </file>
    </personality>
    <block_device_mapping_v2>
        <mapping device_name="/dev/sdb1" source_type="blank" destination_type="local" delete_on_termination="True" guest_format="swap" boot_index="-1"></mapping>
        <mapping device_name="/dev/sda1" source_type="volume" destination_type="volume" uuid="fake-volume-id-1" boot_index="0"></mapping>
    </block_device_mapping_v2>
```

```
</server>
```

## 4.44.2.2. Response

### Example 4.384. Create server: 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"
      }
    ]
  }
}
```

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

Name	Type	Description
security_groups	String <i>(Optional)</i>	A security_groups object.
security_group	String <i>(Optional)</i>	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.
user_data	String <i>(Optional)</i>	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String <i>(Optional)</i>	The availability zone in which to launch the server.
server	ServerFor Create <i>(Required)</i>	server.
imageRef	String <i>(Required)</i>	The image reference for the desired image for your server instance. Specify as an ID or full URL.
flavorRef	String <i>(Required)</i>	The flavor reference for the desired flavor for your server instance. Specify as an ID or full URL.
networks	String <i>(Optional)</i>	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant. Optionally, you can create one or more NICs on the server. To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a networks object. To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a networks object.

Name	Type	Description
		You can specify multiple NICs on the server.
uuid	String (Optional)	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object. Required if you omit the <code>port</code> attribute.
port	String (Optional)	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object. Required if you omit the <code>uuid</code> attribute.
fixed_ip	String (Optional)	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
name	String (Required)	The server name.
metadata	String (Optional)	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
personality	String (Optional)	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit it refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
config_drive	String (Optional)	Enables metadata injection in a server through a configuration drive. To enable a configuration drive, specify <code>true</code> . Otherwise, specify <code>false</code> .

#### Example 4.385. Create server: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" id="fdbce07b-097e-4ab1-8141-b1c847861aa1" adminPass="zA62GVkFvN74">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/fdbce07b-097e-4ab1-8141-b1c847861aa1" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/fdbce07b-097e-4ab1-8141-b1c847861aa1" rel="bookmark"/>
</server>
```

## 4.45. Server OS-EXT-IPS-MAC:mac\_addr extended attribute (servers)

Add OS-EXT-IPS-MAC:mac\_addr extended attribute when you create, show information for, or list servers.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers	Creates a server with the OS-EXT-IPS-MAC:mac_addr extended attribute.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}{?changes-since,image,flavor,name,marker,limit,status,host}	Shows information for a specified server. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.
<b>GET</b>	/v2/{tenant_id}/servers/detail{?changes-since,image,flavor,name,marker,limit,status,host}	Lists details for all servers. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.

## 4.45.1. Create server with OS-EXT-IPS-MAC:mac\_addr extended attribute

Method	URI	Description
POST	/v2/{tenant_id}/servers	Creates a server with the OS-EXT-IPS-MAC:mac_addr extended attribute.

Normal response codes: 202

### 4.45.1.1. Request

This table shows the URI parameters for the create server with os-ext-ips-mac:mac\_addr extended attribute request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

### Example 4.386. Create server with OS-EXT-IPS-MAC:mac\_addr extended attribute: 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": "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3Zlc
yBpbBqdXN0IHN1Y2ggYSBkaXJ1Y3Rpb2
="}
        ]
    }
}
```

This table shows the body parameters for the create server with os-ext-ips-mac:mac\_addr extended attribute request:

Name	Type	Description
security_groups	String (Optional)	A security_groups object.
security_group	String (Optional)	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.
user_data	String (Optional)	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String	The availability zone in which to launch the server.

Name	Type	Description
	(Optional)	
server	ServerFor Create (Required)	server.
imageRef	String (Required)	The image reference for the desired image for your server instance. Specify as an ID or full URL.
flavorRef	String (Required)	The flavor reference for the desired flavor for your server instance. Specify as an ID or full URL.
networks	String (Optional)	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant.  Optionally, you can create one or more NICs on the server.  To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object.  To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object.  You can specify multiple NICs on the server.
uuid	String (Optional)	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object. Required if you omit the <code>port</code> attribute.
port	String (Optional)	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object. Required if you omit the <code>uuid</code> attribute.
fixed_ip	String (Optional)	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
name	String (Required)	The server name.
metadata	String (Optional)	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
personality	String (Optional)	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
config_drive	String (Optional)	Enables metadata injection in a server through a configuration drive. To enable a configuration drive, specify <code>true</code> . Otherwise, specify <code>false</code> .
OS-EXT-IPS-MAC:mac_addr	String (Optional)	Associated MAC address.

#### Example 4.387. Create server with OS-EXT-IPS-MAC:mac\_addr extended attribute: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1" imageRef="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" flavorRef="http://openstack.example.com/openstack/flavors/1" name="new-server-test">
  <metadata>
    <meta key="My Server Name">Apache1</meta>
```

```

</metadata>
<personality>
    <file path="/etc/banner.txt">
        ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
        dcBtb3ZlcYBpbIBqdxN0IHN1Y2ggYSBkaXJ1Y3RpB24gYW5k
        IGF0IHN1Y2ggYSBzcgV1ZC4uLk10IGZ1ZWxzIGFuIGltcHVs
        c2lvbi4uLnRoaXMgaXMgdGh1IHBsYWN1IHRvIGdvIG5vdy4g
        QnV0IHRoZSBza3kga25vd3MgdGh1IHZ1YXNvbnnMgYW5kIHRo
        ZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3VkcwgYW5kIHZ1v
        dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbG1mdCB5b3VY
        c2VsZiBoawdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
        b25zLiINCg0KLVJpY2hhcmQgQmFjaA==
    </file>
</personality>
</server>

```

This operation does not accept a request body.

#### 4.45.1.2. Response

##### Example 4.388. Create server with OS-EXT-IPS-MAC:mac\_addr extended attribute: JSON response

```
{
    "server": {
        "adminPass": "zD7wDKTXiHsp",
        "id": "b44e5008-42f7-4048-b4c8-f40a29da88ba",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/servers/b44e5008-42f7-4048-b4c8-f40a29da88ba",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/b44e5008-42f7-4048-b4c8-f40a29da88ba",
                "rel": "bookmark"
            }
        ]
    }
}
```

##### Example 4.389. Create server with OS-EXT-IPS-MAC:mac\_addr extended attribute: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" id="752dd57d-933b-4a57-a0ae-4c3431c5abc7" adminPass="B2gvFFjBQCvQ">
    <metadata/>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/752dd57d-933b-4a57-a0ae-4c3431c5abc7" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/752dd57d-933b-4a57-a0ae-4c3431c5abc7" rel="bookmark"/>
</server>

```

This operation does not return a response body.

## 4.45.2. Show server information

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}{?changes-since,image,flavor,name,marker,limit,status,host}	Shows information for a specified server. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.

**Normal response codes:** 202

### 4.45.2.1. Request

This table shows the URI parameters for the show server information 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 table shows the query parameters for the show server information 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>	AnyURI	Name of the image in URL format.
flavor <i>(Optional)</i>	AnyURI	Name of the flavor in URL format.
name <i>(Optional)</i>	String	Name of the server as a string.
marker <i>(Optional)</i>	UUID	UUID of the server at which you want to set a marker.
limit <i>(Optional)</i>	Int	Integer value for the limit of values to return.
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.

This operation does not accept a request body.

### 4.45.2.2. Response

#### Example 4.390. Show server information: JSON response

```
{
  "server": {
    "accessIPv4": "",
```

```
"accessIPv6": "",  
"addresses": [  
    "private": [  
        {  
            "addr": "192.168.0.3",  
            "version": 4,  
            "OS-EXT-IPS-MAC:mac_addr": "00:0c:29:e1:42:90"  
        }  
    ]  
],  
"created": "2013-02-07T18:46:28Z",  
"flavor": {  
    "id": "1",  
    "links": [  
        {  
            "href": "http://openstack.example.com/openstack/flavors/  
1",  
            "rel": "bookmark"  
        }  
    ]  
},  
"hostId": "4e2003eddbfdb1280c2618d04090bcdd6773203b8da8347af0b2723d",  
"id": "dc7281f9-ee47-40b9-9950-9f73e7961caa",  
"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/  
dc7281f9-ee47-40b9-9950-9f73e7961caa",  
        "rel": "self"  
    },  
    {  
        "href": "http://openstack.example.com/openstack/servers/  
dc7281f9-ee47-40b9-9950-9f73e7961caa",  
        "rel": "bookmark"  
    }  
],  
"metadata": {  
    "My Server Name": "Apache1"  
},  
"name": "new-server-test",  
"progress": 0,  
"status": "ACTIVE",  
"tenant_id": "openstack",  
"updated": "2013-02-07T18:46:29Z",  
"user_id": "fake"  
}  
}
```

### Example 4.391. Show server information: XML response

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

```
<server xmlns:OS-EXT-IPS-MAC="http://docs.openstack.org/compute/
ext/extended_ips_mac/api/v1.1" xmlns:atom="http://www.w3.
org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/
v1.1" status="ACTIVE" updated="2013-02-07T18:46:29Z" hostId=
"068cc5e2de14b6e533a239c6eac0a0bdedcd57cab25450a6d3da43af" name="new-
server-test" created="2013-02-07T18:46:28Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="22e7cf4d-
ab7a-4a3d-9599-7d0dbaf9ed55">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" addr="192.168.0.3" OS-EXT-IPS-MAC:mac_addr=
"00:0c:29:e1:42:90"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/22e7cf4d-
ab7a-4a3d-9599-7d0dbaf9ed55" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/22e7cf4d-
ab7a-4a3d-9599-7d0dbaf9ed55" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 4.45.3. Get server details

Method	URI	Description
GET	/v2/{tenant_id}/servers/detail{?changes-since,image,flavor,name,marker,limit,status,host}	Lists details for all servers. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.

Normal response codes: 202

### 4.45.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

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

Name	Type	Description
changes-since	DateTime <i>(Optional)</i>	A time/date stamp for when the server last changed status.
image	AnyURI <i>(Optional)</i>	Name of the image in URL format.
flavor	AnyURI <i>(Optional)</i>	Name of the flavor in URL format.
name	String <i>(Optional)</i>	Name of the server as a string.
marker	UUID <i>(Optional)</i>	UUID of the server at which you want to set a marker.
limit	Int <i>(Optional)</i>	Integer value for the limit of values to return.
status	ServerStatus <i>(Optional)</i>	Value of the status of the server so that you can filter on "ACTIVE" for example.
host	String <i>(Optional)</i>	Name of the host as a string.

This operation does not accept a request body.

### 4.45.3.2. Response

#### Example 4.392. Get server details: JSON response

```
{
  "servers": [
    {
      "accessIPv4": "",
      "accessIPv6": "",
      "addresses": {
        "OS-EXT-IPS-MAC:mac_addr": "00:0c:29:7f:1d:4e"
      }
    }
  ]
}
```

```

    "private": [
        {
            "addr": "192.168.0.3",
            "version": 4,
            "OS-EXT-IPS-MAC:mac_addr": "00:0c:29:e1:42:90"
        }
    ],
    "created": "2013-02-07T18:40:59Z",
    "flavor": {
        "id": "1",
        "links": [
            {
                "href": "http://openstack.example.com/openstack/
flavors/1",
                "rel": "bookmark"
            }
        ]
    },
    "hostId":
"fe866a4962fe3bdb6c2db9c8f7dcdb9555aca73387e72b5cb9c45bd3",
    "id": "76908712-653a-4d16-807e-d89d41435d24",
    "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/76908712-653a-4d16-807e-d89d41435d24",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/servers/
76908712-653a-4d16-807e-d89d41435d24",
            "rel": "bookmark"
        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2013-02-07T18:40:59Z",
    "user_id": "fake"
}
]
}

```

### Example 4.393. Get server details: XML response

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

```

<servers xmlns:OS-EXT-IPS-MAC="http://docs.openstack.org/compute/ext/
extended_ips_mac/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2013-02-07T18:40:59Z" hostId=
"51a80e6ee89b638b2cb57eb4e39d89a725e07c8a698f4d8e256f8665" name=
"new-server-test" created="2013-02-07T18:40:59Z" userId="fake"
    tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id=
"0337de6b-1d43-46c8-8804-35669f1dea9a">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" addr="192.168.0.3" OS-EXT-IPS-MAC:mac_addr=
"00:0c:29:e1:42:90"/>
            </network>
        </addresses>
        <atom:link href="http://openstack.example.com/v2/openstack/servers/
0337de6b-1d43-46c8-8804-35669f1dea9a" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/servers/
0337de6b-1d43-46c8-8804-35669f1dea9a" rel="bookmark"/>
    </server>
</servers>
```

This operation does not return a response body.

## 4.46. Configuration drive (servers)

Extend servers and images with a configuration drive.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers	Creates a server with the configuration drive extended attribute.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server including the configuration drive extended attribute.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/detail	Lists details for all servers including the configuration drive extended attribute.

## 4.46.1. Create server with configuration drive

Method	URI	Description
POST	/v2/{tenant_id}/servers	Creates a server with the configuration drive extended attribute.

Normal response codes: 202

### 4.46.1.1. Request

This table shows the URI parameters for the create server with configuration drive request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.394. Create server with configuration drive: 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": "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcyBpbBqdBXN0IHN1Y2ggYSBkaXJ1Y3Rpb2"
            }
        ]
    }
}
```

This table shows the body parameters for the create server with configuration drive request:

Name	Type	Description
security_groups	String (Optional)	A security_groups object.
security_group	String (Optional)	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.
user_data	String (Optional)	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String (Optional)	The availability zone in which to launch the server.
server	ServerFor Create	server.

Name	Type	Description
	(Required)	
imageRef	String	The image reference for the desired image for your server instance.
	(Required)	Specify as an ID or full URL.
flavorRef	String	The flavor reference for the desired flavor for your server instance.
	(Required)	Specify as an ID or full URL.
networks	String (Optional)	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant.  Optionally, you can create one or more NICs on the server.  To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object.  To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object.  You can specify multiple NICs on the server.
uuid	String (Optional)	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object. Required if you omit the <code>port</code> attribute.
port	String (Optional)	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object. Required if you omit the <code>uuid</code> attribute.
fixed_ip	String (Optional)	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
name	String (Required)	The server name.
metadata	String (Optional)	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
personality	String (Optional)	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
config_drive	String (Optional)	Enables metadata injection in a server through a configuration drive. To enable a configuration drive, specify <code>true</code> . Otherwise, specify <code>false</code> .

### Example 4.395. Create server with configuration drive: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1" imageRef="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" flavorRef="http://openstack.example.com/openstack/flavors/1" name="new-server-test">
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
      dCBtb3ZlcycBpbIBqdxN0IHN1Y2ggYSBkaXJ1Y3Rpb24gYW5k
      IGF0IHN1Y2ggYSBzCGV1ZC4uIk10IGZlZWxzIGFuIGltcHVs
      c2lvbi4uLnRoaXMgaXMgdGh1IHBsYWN1IHRvIGdvIG5vdy4g
      QnV0IHRoZSBza3kg25vd3MgdGh1IHZ1YXNvbnnMgYW5kIHRo
    </file>
  </personality>
</server>
```

```

ZSBwYXR0ZXJuicyBiZWhpbmQgYWxsIGNsb3VkcwY5kiH1v
dSB3aWxsIGtub3csIHRvbywd2h1biB5b3UgbG1mdCB5b3VY
c2VsZiBoAwd0IGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
b25zLiINCg0KLVJpY2hhcmQgQmFjaA==
</file>
</personality>
</server>

```

This operation does not accept a request body.

#### 4.46.1.2. Response

##### Example 4.396. Create server with configuration drive: JSON response

```
{
  "server": {
    "adminPass": "am5LKVsBVQ4s",
    "id": "58da039c-dc81-4d8f-8688-a2f819e2f750",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/
58da039c-dc81-4d8f-8688-a2f819e2f750",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/
58da039c-dc81-4d8f-8688-a2f819e2f750",
        "rel": "bookmark"
      }
    ]
  }
}
```

##### Example 4.397. Create server with configuration drive: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1" id="00bba779-f8ae-403e-901d-1af18bcb9187" adminPass=
"XJqvFkH62TZh">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/00bba779-
f8ae-403e-901d-1af18bcb9187" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/00bba779-
f8ae-403e-901d-1af18bcb9187" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 4.46.2. Get server information with configuration drive

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server including the configuration drive extended attribute.

**Normal response codes:** 202

### 4.46.2.1. Request

This table shows the URI parameters for the get server information with configuration drive 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.

### 4.46.2.2. Response

#### Example 4.398. Get server information with configuration drive: JSON response

```
{
  "server": {
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": [
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "config_drive": "",
    "created": "2013-02-04T13:17:50Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "8725fb615b191d8249a40f3e90d1efde88d914412e4edb2719176af",
    "id": "dd3b0715-a3fc-43d8-bbd2-2720beb226fb",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    }
  }
}
```

```

        "rel": "bookmark"
    }
]
},
"links": [
{
    "href": "http://openstack.example.com/v2/openstack/servers/
dd3b0715-a3fc-43d8-bbd2-2720beb226fb",
    "rel": "self"
},
{
    "href": "http://openstack.example.com/openstack/servers/
dd3b0715-a3fc-43d8-bbd2-2720beb226fb",
    "rel": "bookmark"
}
],
"metadata": {
    "My Server Name": "Apache1"
},
"name": "new-server-test",
"progress": 0,
"status": "ACTIVE",
"tenant_id": "openstack",
"updated": "2013-02-04T13:17:51Z",
"user_id": "fake"
}
}
}

```

This table shows the body parameters for the get server information with configuration drive response:

Name	Type	Description
security_groups	String <i>(Optional)</i>	A security_groups object.
security_group	String <i>(Optional)</i>	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.
user_data	String <i>(Optional)</i>	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String <i>(Optional)</i>	The availability zone in which to launch the server.
server	ServerFor Create <i>(Required)</i>	server.
imageRef	String <i>(Required)</i>	The image reference for the desired image for your server instance. Specify as an ID or full URL.
flavorRef	String <i>(Required)</i>	The flavor reference for the desired flavor for your server instance. Specify as an ID or full URL.
networks	String <i>(Optional)</i>	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant.  Optionally, you can create one or more NICs on the server.  To provision the server instance with a NIC for a nova-network network, specify the UUID in the uuid attribute in a networks object.

Name	Type	Description
		To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object.  You can specify multiple NICs on the server.
<code>uuid</code>	String (Optional)	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object. Required if you omit the <code>port</code> attribute.
<code>port</code>	String (Optional)	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object. Required if you omit the <code>uuid</code> attribute.
<code>fixed_ip</code>	String (Optional)	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
<code>name</code>	String (Required)	The server name.
<code>metadata</code>	String (Optional)	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
<code>personality</code>	String (Optional)	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
<code>config_drive</code>	String (Optional)	Enables metadata injection in a server through a configuration drive.  To enable a configuration drive, specify <code>true</code> . Otherwise, specify <code>false</code> .

### Example 4.399. Get server information with configuration drive: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" status="ACTIVE" updated="2013-02-04T13:26:10Z"
  hostId="7a8c3fc15db5d6227d26d5ef559b77c880bbe99da5ce5f5871fc113e"
  name="new-server-test" created="2013-02-04T13:26:09Z" userId="fake"
  tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id=
  "3b9e0572-3d7b-4e6f-9c21-35ad0f7dbf95" config_drive=""
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
      <atom:link href="http://openstack.example.com/openstack/images/
  70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
      <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
  "bookmark"/>
    </flavor>
    <metadata>
      <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
      <network id="private">
        <ip version="4" addr="192.168.0.3"/>
      </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/
  3b9e0572-3d7b-4e6f-9c21-35ad0f7dbf95" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/
  3b9e0572-3d7b-4e6f-9c21-35ad0f7dbf95" rel="bookmark"/>
  </server>
```

This operation does not return a response body.

## 4.46.3. Get server details with configuration drive

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/detail	Lists details for all servers including the configuration drive extended attribute.

**Normal response codes:** 202

### 4.46.3.1. Request

This table shows the URI parameters for the get server details with configuration drive 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.

### 4.46.3.2. Response

#### Example 4.400. Get server details with configuration drive: JSON response

```
{
    "servers": [
        {
            "accessIPv4": "",
            "accessIPv6": "",
            "addresses": [
                "private": [
                    {
                        "addr": "192.168.0.3",
                        "version": 4
                    }
                ]
            ],
            "config_drive": "",
            "created": "2013-02-04T13:21:44Z",
            "flavor": {
                "id": "1",
                "links": [
                    {
                        "href": "http://openstack.example.com/openstack/
flavors/1",
                        "rel": "bookmark"
                    }
                ]
            },
            "hostId": "76e154b0015e25fad65a7ab0c35a86dd79acfa8312075a6534ef6176",
            "id": "720e688f-5ec8-4d4f-b585-dbd1a89ceeb0",
            "image": {
                "minDisk": 0,
                "minMemory": 256,
                "name": "Ubuntu 12.04 LTS (Precise Pangolin) - Cloud Image (v1.0)"}
```

```

        "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/720e688f-5ec8-4d4f-b585-dbd1a89ceeb0",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
720e688f-5ec8-4d4f-b585-dbd1a89ceeb0",
                "rel": "bookmark"
            }
        ],
        "metadata": {
            "My Server Name": "Apache1"
        },
        "name": "new-server-test",
        "progress": 0,
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2013-02-04T13:21:44Z",
        "user_id": "fake"
    }
]
}

```

This table shows the body parameters for the get server details with configuration drive response:

Name	Type	Description
security_groups	String <i>(Optional)</i>	A security_groups object.
security_group	String <i>(Optional)</i>	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.
user_data	String <i>(Optional)</i>	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String <i>(Optional)</i>	The availability zone in which to launch the server.
server	ServerFor Create <i>(Required)</i>	server.
imageRef	String <i>(Required)</i>	The image reference for the desired image for your server instance. Specify as an ID or full URL.
flavorRef	String <i>(Required)</i>	The flavor reference for the desired flavor for your server instance. Specify as an ID or full URL.

Name	Type	Description
networks	String <i>(Optional)</i>	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant.  Optionally, you can create one or more NICs on the server.  To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object.  To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object.  You can specify multiple NICs on the server.
uuid	String <i>(Optional)</i>	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object. Required if you omit the <code>port</code> attribute.
port	String <i>(Optional)</i>	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object. Required if you omit the <code>uuid</code> attribute.
fixed_ip	String <i>(Optional)</i>	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
name	String <i>(Required)</i>	The server name.
metadata	String <i>(Optional)</i>	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
personality	String <i>(Optional)</i>	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
config_drive	String <i>(Optional)</i>	Enables metadata injection in a server through a configuration drive.  To enable a configuration drive, specify <code>true</code> . Otherwise, specify <code>false</code> .

### Example 4.401. Get server details with configuration drive: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2013-02-04T13:26:27Z" hostId="2a00edcff768661880eb9c96c951f56c2c5dcd873bb652361008efc7" name="new-server-test" created="2013-02-04T13:26:27Z" userId="fake" tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id="515d94d3-aae4-4bd5-bb4e-9601c657372f" config_drive="">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/openstack/flavors/1" rel="bookmark"/>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" addr="192.168.0.3" />
            </network>
        </addresses>
    </server>
</servers>
```

```
</network>
</addresses>
<atom:link href="http://openstack.example.com/v2/openstack/servers/
515d94d3-aee4-4bd5-bb4e-9601c657372f" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/servers/515d94d3-
aee4-4bd5-bb4e-9601c657372f" rel="bookmark"/>
</server>
</servers>
```

This operation does not return a response body.

## 4.47. Servers with extended availability zones (servers)

Show the instance availability zone for compute nodes (nova-compute). Internal services appear in their own *internal* availability zone.

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server, including its availability zone.
GET	/v2/{tenant_id}/servers/detail	Lists details for servers, including their current availability zone.

## 4.47.1. Show server

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server, including its availability zone.

**Normal response codes:** 200

### 4.47.1.1. Request

This table shows the URI parameters for the show server 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.

### 4.47.1.2. Response

#### Example 4.402. Show server: JSON response

```
{
    "server": {
        "OS-EXT-AZ:availability_zone": "nova",
        "accessIPv4": "",
        "accessIPv6": "",
        "addresses": {
            "private": [
                {
                    "addr": "192.168.0.3",
                    "version": 4
                }
            ]
        },
        "created": "2013-01-30T13:38:47Z",
        "flavor": {
            "id": "1",
            "links": [
                {
                    "href": "http://openstack.example.com/openstack/flavors/1",
                    "rel": "bookmark"
                }
            ]
        },
        "hostId": "d38ea49a033b0efaf80c165de63f4805c886dfb94dc0fe731227eccb",
        "id": "fb7babfd-e1a1-4add-90e6-3558180983c7",
        "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/
fb7babfd-e1a1-4add-90e6-3558180983c7",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/servers/
fb7babfd-e1a1-4add-90e6-3558180983c7",
            "rel": "bookmark"
        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2013-01-30T13:38:49Z",
    "user_id": "fake"
}
}
}

```

### Example 4.403. Show server: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-EXT-AZ="http://docs.openstack.org/compute/ext/
extended_availability_zone/api/v2" xmlns:atom="http://www.w3.
org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/
v1.1" status="ACTIVE" updated="2013-01-30T14:29:20Z" hostId=
"471e52951e3182954c5a93489dafc3fc38a9ef3e0b62d26dc740460c" name="new-
server-test" created="2013-01-30T14:29:19Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="26ea8424-758d-483a-
addc-9a5905afc9e6" OS-EXT-AZ:availability_zone="nova">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" addr="192.168.0.3"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/
26ea8424-758d-483a-addc-9a5905afc9e6" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/
26ea8424-758d-483a-addc-9a5905afc9e6" rel="bookmark"/>
</server>

```

This operation does not return a response body.

## 4.47.2. List details for servers

Method	URI	Description
GET	/v2/{tenant_id}/servers/detail	Lists details for servers, including their current availability zone.

Normal response codes: 200

### 4.47.2.1. Request

This table shows the URI parameters for the list details for servers 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.

### 4.47.2.2. Response

#### Example 4.404. List details for servers: JSON response

```
{
  "servers": [
    {
      "OS-EXT-AZ:availability_zone": "nova",
      "accessIPv4": "",
      "accessIPv6": "",
      "addresses": [
        "private": [
          {
            "addr": "192.168.0.3",
            "version": 4
          }
        ],
        "created": "2013-01-30T13:26:51Z",
        "flavor": {
          "id": "1",
          "links": [
            {
              "href": "http://openstack.example.com/openstack/
flavors/1",
              "rel": "bookmark"
            }
          ]
        },
        "hostId": "60c988a84401fa15888a32833e5848e9caa99a45778310ba7b363165",
        "id": "3dbf5b00-dabc-41ff-b6ab-4409568fae9d",
        "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/3dbf5b00-dabc-41ff-b6ab-4409568fae9d",
    "rel": "self"
},
{
    "href": "http://openstack.example.com/openstack/servers/
3dbf5b00-dabc-41ff-b6ab-4409568fae9d",
    "rel": "bookmark"
},
],
"metadata": {
    "My Server Name": "Apache1"
},
"name": "new-server-test",
"progress": 0,
"status": "ACTIVE",
"tenant_id": "openstack",
"updated": "2013-01-30T13:26:52Z",
"user_id": "fake"
}
]
}
}

```

### Example 4.405. List details for servers: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:OS-EXT-AZ="http://docs.openstack.org/compute/ext/
extended_availability_zone/api/v2" xmlns:atom="http://www.w3.org/2005/Atom"
xmlns="http://docs.openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2013-01-30T14:29:20Z" hostId=
"85adf7d0492dedf0a7e3dc44ef7d16186b768ca3df33c4d608e630d9" name="new-server-
test" created="2013-01-30T14:29:19Z" userId="fake" tenantId="openstack"
accessIPv4="" accessIPv6="" progress="0" id="a668c72d-2bac-4806-a297-
c7c11d97e3b3" OS-EXT-AZ:availability_zone="nova">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" addr="192.168.0.3"/>
            </network>
        </addresses>
        <atom:link href="http://openstack.example.com/v2/openstack/servers/
a668c72d-2bac-4806-a297-c7c11d97e3b3" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/servers/
a668c72d-2bac-4806-a297-c7c11d97e3b3" rel="bookmark"/>
    </server>

```

```
</server>
</servers>
```

This operation does not return a response body.

## 4.48. Servers and images with disk config (servers, images)

Extend servers with the diskConfig attribute.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers	Creates a server.
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server.
<b>PUT</b>	/v2/{tenant_id}/servers/{server_id}	Updates a specified server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Resizes a server.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Rebuilds a specified server.
<b>GET</b>	/v2/{tenant_id}/servers/detail	Lists servers.
<b>GET</b>	/v2/{tenant_id}/images/{image_id}	Gets information for a specified image.
<b>GET</b>	/v2/{tenant_id}/images/detail	Lists images.

## 4.48.1. Create server

Method	URI	Description
POST	/v2/{tenant_id}/servers	Creates a server.

**Normal response codes:** 202

### 4.48.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.406. Create server: 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":
                    "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3Zlc
                    yBpbIBqdXN0IHN1Y2ggYSBkaXJ1Y3RpB2
                    ="
            }
        ]
    }
}
```

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

Name	Type	Description
security_groups	String <i>(Optional)</i>	A security_groups object.
security_group	String <i>(Optional)</i>	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.
user_data	String <i>(Optional)</i>	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String <i>(Optional)</i>	The availability zone in which to launch the server.
server	ServerFor Create <i>(Required)</i>	server.
imageRef	String	The image reference for the desired image for your server instance.

Name	Type	Description
	(Required)	Specify as an ID or full URL.
flavorRef	String	The flavor reference for the desired flavor for your server instance.
	(Required)	Specify as an ID or full URL.
networks	String (Optional)	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant.  Optionally, you can create one or more NICs on the server.  To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object.  To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object.  You can specify multiple NICs on the server.
uuid	String (Optional)	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object. Required if you omit the <code>port</code> attribute.
port	String (Optional)	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object. Required if you omit the <code>uuid</code> attribute.
fixed_ip	String (Optional)	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
name	String (Required)	The server name.
metadata	String (Optional)	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
personality	String (Optional)	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
config_drive	String (Optional)	Enables metadata injection in a server through a configuration drive.  To enable a configuration drive, specify <code>true</code> . Otherwise, specify <code>false</code> .
OS-DCF:diskConfig	(Optional)	A valid value is AUTO or MANUAL.

### Example 4.407. Create server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1" imageRef="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" flavorRef="http://openstack.example.com/openstack/flavors/1" name="new-server-test">
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
      dCBtb3ZlcycBpbBqdXN0IHN1Y2ggYSBkaXJ1Y3RpB24gYW5k
      IGF0IHN1Y2ggYSBzCGV1ZC4uLk10IGZlZWxzIGFuIGltcHVs
      c2lvbi4uLnRoaXMgaXMgdGh1IHByYWN1IHRvIGdvIG5vdy4g
      QnV0IHRoZSBza3kg25vd3MgdGh1IHJ1YXNvbMgYW5kIHRo
      ZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3VkcwgYW5kIHLv
      dSB3aWxsIGtub3csIHRvbywd2hlbiB5b3UgbGlmdCB5b3Vy
    </file>
  </personality>
</server>
```

```
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
b25zLiINCg0KLVJpY2hhcmQgQmFjaA==
</file>
</personality>
</server>
```

This operation does not accept a request body.

#### 4.48.1.2. Response

##### Example 4.408. Create server: JSON response

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "adminPass": "CQH9gWzgkVno",
    "id": "324dfb7d-f4a9-419a-9a19-237df04b443b",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/
324dfb7d-f4a9-419a-9a19-237df04b443b",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/
324dfb7d-f4a9-419a-9a19-237df04b443b",
        "rel": "bookmark"
      }
    ]
  }
}
```

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

Name	Type	Description
security_groups <i>(Optional)</i>	String	A security_groups object.
security_group <i>(Optional)</i>	String	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.
user_data <i>(Optional)</i>	String	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone <i>(Optional)</i>	String	The availability zone in which to launch the server.
server <i>(Required)</i>	ServerForCreate	server.
imageRef <i>(Required)</i>	String	The image reference for the desired image for your server instance. Specify as an ID or full URL.
flavorRef <i>(Required)</i>	String	The flavor reference for the desired flavor for your server instance. Specify as an ID or full URL.
networks	String	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant.

Name	Type	Description
	(Optional)	<p>Optionally, you can create one or more NICs on the server.</p> <p>To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object.</p> <p>To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object.</p> <p>You can specify multiple NICs on the server.</p>
<code>uuid</code>	String (Optional)	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object. Required if you omit the <code>port</code> attribute.
<code>port</code>	String (Optional)	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object. Required if you omit the <code>uuid</code> attribute.
<code>fixed_ip</code>	String (Optional)	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
<code>name</code>	String (Required)	The server name.
<code>metadata</code>	String (Optional)	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
<code>personality</code>	String (Optional)	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
<code>config_drive</code>	String (Optional)	Enables metadata injection in a server through a configuration drive. To enable a configuration drive, specify <code>true</code> . Otherwise, specify <code>false</code> .

### Example 4.409. Create server: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" id="900a4ef7-f374-413f-8816-52d3dbfaf498" adminPass="Cj6sZgWq85qm" OS-DCF:diskConfig="AUTO">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/900a4ef7-f374-413f-8816-52d3dbfaf498" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/900a4ef7-f374-413f-8816-52d3dbfaf498" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 4.48.2. Show server information

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server.

**Normal response codes:** 200

### 4.48.2.1. Request

This table shows the URI parameters for the show server information 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.

### 4.48.2.2. Response

#### Example 4.410. Show server information: JSON response

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": [
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    ],
    "created": "2012-12-02T02:11:55Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "c949ab4256cea23b6089b710aa2df48bf6577ed915278b62e33ad8bb",
    "id": "5046e2f2-3b33-4041-b3cf-e085f73e78e7",
    "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/
5046e2f2-3b33-4041-b3cf-e085f73e78e7",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/servers/
5046e2f2-3b33-4041-b3cf-e085f73e78e7",
            "rel": "bookmark"
        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2012-12-02T02:11:55Z",
    "user_id": "fake"
}
}
}

```

This table shows the body parameters for the show server information response:

Name	Type	Description
OS-DCF:diskConfig	String <i>(Optional)</i>	Valid value is AUTO or MANUAL.

### Example 4.411. Show server information: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/
api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1" status="ACTIVE" updated="2012-12-02T02:15:37Z"
hostId="afa0a883de4743c7a0c164327bda5284b875c50e1a9e30de910ac126" name=
"new-server-test" created="2012-12-02T02:15:37Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="d5d844c0-
ecc3-4202-90ed-2e85b7fe513d" OS-DCF:diskConfig="AUTO">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" addr="192.168.0.3"/>
        </network>
    </addresses>

```

```
<atom:link href="http://openstack.example.com/v2/openstack/servers/d5d844c0-
ecc3-4202-90ed-2e85b7fe513d" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/servers/d5d844c0-
ecc3-4202-90ed-2e85b7fe513d" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 4.48.3. Update server

Method	URI	Description
PUT	/v2/{tenant_id}/servers/{server_id}	Updates a specified server.

Normal response codes: 200

### 4.48.3.1. Request

This table shows the URI parameters for the update server 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 4.412. Update server: JSON request

```
{
    "server": {
        "OS-DCF:diskConfig": "AUTO"
    }
}
```

#### Example 4.413. Update server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
         xmlns:atom="http://www.w3.org/2005/Atom"
         xmlns="http://docs.openstack.org/compute/api/v1.1"
         OS-DCF:diskConfig="AUTO" />
```

This operation does not accept a request body.

### 4.48.3.2. Response

#### Example 4.414. Update server: JSON response

```
{
    "server": {
        "OS-DCF:diskConfig": "AUTO",
        "accessIPv4": "",
        "accessIPv6": "",
        "addresses": [
            "private": [
                {
                    "addr": "192.168.0.3",
                    "version": 4
                }
            ]
        ],
        "created": "2012-12-02T02:11:57Z",
        "flavor": {
            "id": "2",
            "name": "m1.tiny"
        }
    }
}
```

```

        "id": "1",
        "links": [
            {
                "href": "http://openstack.example.com/openstack/flavors/
1",
                "rel": "bookmark"
            }
        ],
        "hostId": "6e84af987b4e7ec1c039b16d21f508f4a505672bd94fb0218b668d07",
        "id": "324dfb7d-f4a9-419a-9a19-237df04b443b",
        "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/
324dfb7d-f4a9-419a-9a19-237df04b443b",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
324dfb7d-f4a9-419a-9a19-237df04b443b",
                "rel": "bookmark"
            }
        ],
        "metadata": {
            "My Server Name": "Apache1"
        },
        "name": "new-server-test",
        "progress": 0,
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2012-12-02T02:11:58Z",
        "user_id": "fake"
    }
}

```

### Example 4.415. Update server: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/
api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1" status="ACTIVE" updated="2012-12-02T02:15:40Z"
hostId="e987bebfb62599c59c4559b249d0f5f300a302d1e9ff22dd1f0c5c4b0" name=
"new-server-test" created="2012-12-02T02:15:40Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="900a4ef7-
f374-413f-8816-52d3dbfaf498" OS-DCF:diskConfig="AUTO">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">

```

```
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel="bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="private">
      <ip version="4" addr="192.168.0.3"/>
    </network>
  </addresses>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/900a4ef7-f374-413f-8816-52d3dbfaf498" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/900a4ef7-f374-413f-8816-52d3dbfaf498" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 4.48.4. Resize server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Resizes a server.

**Normal response codes:** 202

### 4.48.4.1. Request

This table shows the URI parameters for the resize server 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 4.416. Resize server: JSON request

```
{
  "resize": {
    "flavorRef": "3",
    "OS-DCF:diskConfig": "AUTO"
  }
}
```

#### Example 4.417. Resize server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<resize xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
         xmlns:atom="http://www.w3.org/2005/Atom"
         xmlns="http://docs.openstack.org/compute/api/v1.1"
         OS-DCF:diskConfig="AUTO"
         flavorRef="3" />
```

This operation does not accept a request body.

## 4.48.5. Rebuild server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Rebuilds a specified server.

Normal response codes: 202

### 4.48.5.1. Request

This table shows the URI parameters for the rebuild server 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 4.418. Rebuild server: JSON request

```
{
  "rebuild": {
    "imageRef": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
    "OS-DCF:diskConfig": "AUTO"
  }
}
```

#### Example 4.419. Rebuild server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<rebuild xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/
v1.1"
          xmlns:atom="http://www.w3.org/2005/Atom"
          xmlns="http://docs.openstack.org/compute/api/v1.1"
          imageRef="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b"
          OS-DCF:diskConfig="AUTO" />
```

This operation does not accept a request body.

### 4.48.5.2. Response

#### Example 4.420. Rebuild server: JSON response

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    }
  }
}
```

```

        ],
      },
      "adminPass": "NBjMaJoFL4EF",
      "created": "2012-12-02T02:11:56Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "hostId": "c076393ad900d62c4805a42df10d9b364f629842681c00cce035487f",
      "id": "63a8aa13-60fe-41c4-b079-77f6fdf3c841",
      "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/63a8aa13-60fe-41c4-b079-77f6fdf3c841",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/servers/63a8aa13-60fe-41c4-b079-77f6fdf3c841",
          "rel": "bookmark"
        }
      ],
      "metadata": {
        "My Server Name": "Apache1"
      },
      "name": "new-server-test",
      "progress": 0,
      "status": "ACTIVE",
      "tenant_id": "openstack",
      "updated": "2012-12-02T02:11:56Z",
      "user_id": "fake"
    }
  }
}

```

### Example 4.421. Rebuild server: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/
api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1" status="ACTIVE" updated="2012-12-02T02:15:39Z"
hostId="981de784ae4d8c49ca075024977828a16e7f3c2beeb19115b0366e17" name=
"new-server-test" created="2012-12-02T02:15:38Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="10791a94-8900-4d0c-
b93d-0debb224882e" adminPass="mTxoVD3eALpv" OS-DCF:diskConfig="AUTO">

```

```
<image id="70a599e0-31e7-49b7-b260-868f441e862b">
  <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
</image>
<flavor id="1">
  <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
</flavor>
<metadata>
  <meta key="My Server Name">Apache1</meta>
</metadata>
<addresses>
  <network id="private">
    <ip version="4" addr="192.168.0.3"/>
  </network>
</addresses>
<atom:link href="http://openstack.example.com/v2/openstack/servers/
10791a94-8900-4d0c-b93d-0debb224882e" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/servers/
10791a94-8900-4d0c-b93d-0debb224882e" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 4.48.6. List servers

Method	URI	Description
GET	/v2/{tenant_id}/servers/detail	Lists servers.

**Normal response codes:** 200

### 4.48.6.1. Request

This table shows the URI parameters for the list servers 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.

### 4.48.6.2. Response

#### Example 4.422. List servers: JSON response

```
{
    "servers": [
        {
            "OS-DCF:diskConfig": "AUTO",
            "accessIPv4": "",
            "accessIPv6": "",
            "addresses": {
                "private": [
                    {
                        "addr": "192.168.0.3",
                        "version": 4
                    }
                ]
            },
            "created": "2012-12-02T02:11:55Z",
            "flavor": {
                "id": "1",
                "links": [
                    {
                        "href": "http://openstack.example.com/openstack/
flavors/1",
                        "rel": "bookmark"
                    }
                ]
            },
            "hostId": "99428f32351a5d89d0f7727c6eec68c1777c545a0972aaac645508dc",
            "id": "05372e62-05b9-4ee2-9343-9a1fdf2a5fda",
            "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/05372e62-05b9-4ee2-9343-9a1fdf2a5fda",
    "rel": "self"
},
{
    "href": "http://openstack.example.com/openstack/servers/
05372e62-05b9-4ee2-9343-9a1fdf2a5fda",
    "rel": "bookmark"
}
],
"metadata": {
    "My Server Name": "Apache1"
},
"name": "new-server-test",
"progress": 0,
"status": "ACTIVE",
"tenant_id": "openstack",
"updated": "2012-12-02T02:11:56Z",
"user_id": "fake"
}
]
}
}

```

This table shows the body parameters for the list servers response:

Name	Type	Description
OS-DCF:diskConfig	String <i>(Optional)</i>	Valid value is AUTO or MANUAL.

### Example 4.423. List servers: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/
v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1">
    <server status="ACTIVE" updated="2012-12-02T02:15:38Z" hostId=
"85973b57730e91f4eea36b3e7a2a7e3fdaf56008af335dd59f897a59" name=
"new-server-test" created="2012-12-02T02:15:38Z" userId="fake"
    tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id=
"08266bed-2651-4b6c-9dc8-83f0c3ef9d38" OS-DCF:diskConfig="AUTO">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" addr="192.168.0.3" />

```

```
    </network>
  </addresses>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/
08266bed-2651-4b6c-9dc8-83f0c3ef9d38" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/
08266bed-2651-4b6c-9dc8-83f0c3ef9d38" rel="bookmark"/>
</server>
</servers>
```

This operation does not return a response body.

## 4.48.7. Get image information

Method	URI	Description
GET	/v2/{tenant_id}/images/{image_id}	Gets information for a specified image.

**Normal response codes:** 200

### 4.48.7.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{image_id}	String	The UUID for the image of interest to you.

This operation does not accept a request body.

### 4.48.7.2. Response

#### Example 4.424. Get image information: JSON response

```
{
  "image": {
    "OS-DCF:diskConfig": "AUTO",
    "created": "2011-01-01T01:02:03Z",
    "id": "70a599e0-31e7-49b7-b260-868f441e862b",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "bookmark"
      },
      {
        "href": "http://glance.openstack.example.com/openstack/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"
  }
}
```

```
        "updated": "2011-01-01T01:02:03Z"
    }
}
```

This table shows the body parameters for the get image information response:

Name	Type	Description
OS-DCF:diskConfig	String <i>(Optional)</i>	Valid value is AUTO or MANUAL.

### Example 4.425. Get image information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<image xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage7" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="70a599e0-31e7-49b7-b260-868f441e862b" OS-DCF:diskConfig="AUTO">
    <metadata>
        <meta key="kernel_id">nokernel</meta>
        <meta key="auto_disk_config">True</meta>
        <meta key="ramdisk_id">nokernel</meta>
        <meta key="architecture">x86_64</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" type="application/vnd.openstack.image" rel="alternate"/>
</image>
```

This operation does not return a response body.

## 4.48.8. List images

Method	URI	Description
GET	/v2/{tenant_id}/images/detail	Lists images.

**Normal response codes:** 200

### 4.48.8.1. Request

This table shows the URI parameters for the list images 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.

### 4.48.8.2. Response

#### Example 4.426. List images: JSON response

```
{
  "images": [
    {
      "OS-DCF:diskConfig": "AUTO",
      "created": "2011-01-01T01:02:03Z",
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        },
        {
          "href": "http://glance.openstack.example.com/openstack/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"
    }
  ]
}
```

```
{  
    "created": "2011-01-01T01:02:03Z",  
    "id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",  
    "links": [  
        {  
            "href": "http://openstack.example.com/v2/openstack/images/  
155d900f-4e14-4e4c-a73d-069cbf4541e6",  
            "rel": "self"  
        },  
        {  
            "href": "http://openstack.example.com/openstack/images/  
155d900f-4e14-4e4c-a73d-069cbf4541e6",  
            "rel": "bookmark"  
        },  
        {  
            "href": "http://glance.openstack.example.com/openstack/  
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/openstack/images/  
a2459075-d96c-40d5-893e-577ff92e721c",  
            "rel": "self"  
        },  
        {  
            "href": "http://openstack.example.com/openstack/images/  
a2459075-d96c-40d5-893e-577ff92e721c",  
            "rel": "bookmark"  
        },  
        {  
            "href": "http://glance.openstack.example.com/openstack/  
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"
    },
    {
        "OS-DCF:diskConfig": "MANUAL",
        "created": "2011-01-01T01:02:03Z",
        "id": "a440c04b-79fa-479c-bed1-0b816eaec379",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/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/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/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/openstack/images/cedef40a-ed67-4d10-800e-17455edce175",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/cedef40a-ed67-4d10-800e-17455edce175",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/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/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                "rel": "bookmark"
            },
            {

```

```

        "href": "http://glance.openstack.example.com/openstack/
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 table shows the body parameters for the list images response:

Name	Type	Description
OS-DCF:diskConfig	String <i>(Optional)</i>	Valid value is AUTO or MANUAL.

### Example 4.427. List images: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<images xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/
v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1">
    <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage7"
    created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
    "70a599e0-31e7-49b7-b260-868f441e862b" OS-DCF:diskConfig="AUTO">
        <metadata>
            <meta key="kernel_id">nokernel</meta>
            <meta key="auto_disk_config">True</meta>
            <meta key="ramdisk_id">nokernel</meta>
            <meta key="architecture">x86_64</meta>
        </metadata>
        <atom:link href="http://openstack.example.com/v2/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        <atom:link href="http://glance.openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" type="application/vnd.openstack.image"
        rel="alternate"/>
    </image>
    <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
    created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
    "155d900f-4e14-4e4c-a73d-069cbf4541e6">
        <metadata>
            <meta key="kernel_id">nokernel</meta>
            <meta key="ramdisk_id">nokernel</meta>
            <meta key="architecture">x86_64</meta>
        </metadata>
        <atom:link href="http://openstack.example.com/v2/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="self"/>

```

```
<atom:link href="http://openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="a2459075-d96c-40d5-893e-577ff92e721c">
<metadata>
<meta key="kernel_id">nokernel</meta>
<meta key="ramdisk_id">nokernel</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage6" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="a440c04b-79fa-479c-bed1-0b816eaec379" OS-DCF:diskConfig="MANUAL">
<metadata>
<meta key="kernel_id">nokernel</meta>
<meta key="auto_disk_config">False</meta>
<meta key="ramdisk_id">nokernel</meta>
<meta key="architecture">x86_64</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eaec379" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="c905cedb-7281-47e4-8a62-f26bc5fc4c77">
<metadata>
<meta key="kernel_id">155d900f-4e14-4e4c-a73d-069cbf4541e6</meta>
<meta key="ramdisk_id">None</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77" type="application/vnd.openstack.image" rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456" created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id="cedef40a-ed67-4d10-800e-17455edce175">
<metadata>
<meta key="kernel_id">nokernel</meta>
<meta key="ramdisk_id">nokernel</meta>
</metadata>
```

```

<atom:link href="http://openstack.example.com/v2/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/cedef40a-
ed67-4d10-800e-17455edce175" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175" type="application/vnd.openstack.image"
rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"76fa36fc-c930-4bf3-8c8a-ea2a2420deb6">
<metadata>
<meta key="kernel_id">nokernel</meta>
<meta key="ramdisk_id">nokernel</meta>
</metadata>
<atom:link href="http://openstack.example.com/v2/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/images/76fa36fc-
c930-4bf3-8c8a-ea2a2420deb6" rel="bookmark"/>
<atom:link href="http://glance.openstack.example.com/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" type="application/vnd.openstack.image"
rel="alternate"/>
</image>
</images>
```

This operation does not return a response body.

## 4.49. Server IP type (servers)

Show the type of the IP addresses assigned to an instance. Type is either fixed or floating.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}/action	Shows the type of IP assigned to a specified server, either fixed or floating.
<b>GET</b>	/v2/{tenant_id}/servers/detail	Lists servers and shows their IPs by type. Type is either fixed or floating.

## 4.49.1. Show IP type

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/action	Shows the type of IP assigned to a specified server, either fixed or floating.

**Normal response codes:** 200

### 4.49.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{server_id}	UUID	The UUID for the server of interest to you.

This operation does not accept a request body.

### 4.49.1.2. Response

#### Example 4.428. Show IP type: JSON response

```
{
  "server": {
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "OS-EXT-IPS:type": "fixed",
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "created": "2013-02-07T18:46:28Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "4e2003eddbfdb1280c2618d04090bcdd6773203b8da8347af0b2723d",
    "id": "dc7281f9-ee47-40b9-9950-9f73e7961caa",
    "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/dc7281f9-ee47-40b9-9950-9f73e7961caa",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/servers/dc7281f9-ee47-40b9-9950-9f73e7961caa",
            "rel": "bookmark"
        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2013-02-07T18:46:29Z",
    "user_id": "fake"
}
}
```

### Example 4.429. Show IP type: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-EXT-IPS="http://docs.openstack.org/compute/ext/extended_ips/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" status="ACTIVE" updated="2013-02-07T18:46:29Z" hostId="068cc5e2de14b6e533a239c6eac0a0bdedcd57cab25450a6d3da43af" name="new-server-test" created="2013-02-07T18:46:28Z" userId="fake" tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id="22e7cf4d-ab7a-4a3d-9599-7d0dbaf9ed55">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel="bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip OS-EXT-IPS:type="fixed" version="4" addr="192.168.0.3"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/22e7cf4d-ab7a-4a3d-9599-7d0dbaf9ed55" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/22e7cf4d-ab7a-4a3d-9599-7d0dbaf9ed55" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 4.49.2. List servers with IP type

Method	URI	Description
GET	/v2/{tenant_id}/servers/detail	Lists servers and shows their IPs by type. Type is either fixed or floating.

Normal response codes: 200

### 4.49.2.1. Request

This table shows the URI parameters for the list servers with ip type request:

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

This operation does not accept a request body.

### 4.49.2.2. Response

#### Example 4.430. List servers with IP type: JSON response

```
{
  "servers": [
    {
      "accessIPv4": "",
      "accessIPv6": "",
      "addresses": {
        "private": [
          {
            "OS-EXT-IPS:type": "fixed",
            "addr": "192.168.0.3",
            "version": 4
          }
        ]
      },
      "created": "2013-02-07T18:40:59Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "hostId": "fe866a4962fe3bdb6c2db9c8f7dcdb9555aca73387e72b5cb9c45bd3",
      "id": "76908712-653a-4d16-807e-d89d41435d24",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
images/70a599e0-31e7-49b7-b260-868f441e862b",
            "rel": "bookmark"
          }
        ]
      }
    }
  ]
}
```

```

        "rel": "bookmark"
    }
]
},
"links": [
{
    "href": "http://openstack.example.com/v2/openstack/
servers/76908712-653a-4d16-807e-d89d41435d24",
    "rel": "self"
},
{
    "href": "http://openstack.example.com/openstack/servers/
76908712-653a-4d16-807e-d89d41435d24",
    "rel": "bookmark"
}
],
"metadata": {
    "My Server Name": "Apache1"
},
"name": "new-server-test",
"progress": 0,
"status": "ACTIVE",
"tenant_id": "openstack",
"updated": "2013-02-07T18:40:59Z",
"user_id": "fake"
}
]
}
}

```

### Example 4.431. List servers with IP type: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:OS-EXT-IPS="http://docs.openstack.org/compute/ext/extended_ips/
api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2013-02-07T18:40:59Z" hostId=
"51a80e6ee89b638b2cb57eb4e39d89a725e07c8a698f4d8e256f8665" name=
"new-server-test" created="2013-02-07T18:40:59Z" userId="fake"
    tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id=
"0337de6b-1d43-46c8-8804-35669f1dea9a">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip OS-EXT-IPS:type="fixed" version="4" addr="192.168.0.3" />
            </network>
        </addresses>
        <atom:link href="http://openstack.example.com/v2/openstack/servers/
0337de6b-1d43-46c8-8804-35669f1dea9a" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/servers/
0337de6b-1d43-46c8-8804-35669f1dea9a" rel="bookmark"/>
    </server>

```

```
</server>
</servers>
```

This operation does not return a response body.

## 4.50. Server extended attributes (servers)

Show metadata for servers.

Method	URI	Description
<b>GET</b>	/v2/servers	Lists detailed extended server attribute information for all servers.
<b>GET</b>	/v2/servers/{server_id}	Shows extended server attributes for a specified server.

## 4.50.1. List servers with extended server attributes

Method	URI	Description
GET	/v2/servers	Lists detailed extended server attribute information for all servers.

Normal response codes: 200

### 4.50.1.1. Request

This operation does not accept a request body.

### 4.50.1.2. Response

#### Example 4.432. List servers with extended server attributes: JSON response

```
{
  "servers": [
    {
      "OS-EXT-SRV-ATTR:host": "dd99797793774612b081a8be19bf721a",
      "OS-EXT-SRV-ATTR:hypervisor_hostname": "fake-mini",
      "OS-EXT-SRV-ATTR:instance_name": "instance-00000001",
      "accessIPv4": "",
      "accessIPv6": "",
      "addresses": [
        "private": [
          {
            "addr": "192.168.0.3",
            "version": 4
          }
        ]
      ],
      "created": "2012-11-15T19:27:05Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "hostId": "146245c049213a54b8c2352751518fcf4c2befd1b942b45a5a705d35",
      "id": "e0c3563a-84ef-4d0b-bb80-23392cd23882",
      "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/e0c3563a-84ef-4d0b-bb80-23392cd23882",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
e0c3563a-84ef-4d0b-bb80-23392cd23882",
                "rel": "bookmark"
            }
        ],
        "metadata": {
            "My Server Name": "Apache1"
        },
        "name": "new-server-test",
        "progress": 0,
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2012-11-15T19:27:05Z",
        "user_id": "fake"
    }
]
}

```

### Example 4.433. List servers with extended server attributes: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:OS-EXT-SRV-ATTR="http://docs.openstack.org/compute/ext/
extended_status/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2012-11-15T19:27:06Z" hostId=
"b348a7376e2e61781829c9b45e63675aa0207632c25ce36c55a4fb2a" name="new-
server-test" created="2012-11-15T19:27:06Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="3cadb9e9-
f430-4f62-8b9b-3efb671ff1fa" OS-EXT-SRV-ATTR:instance_name="instance-00000001"
OS-EXT-SRV-ATTR:host="2c4d049170fe409abc14942757d63a4e" OS-EXT-SRV-
ATTR:hypervisor_hostname="fake-mini">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark" />
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" addr="192.168.0.3" />
            </network>
        </addresses>
        <atom:link href="http://openstack.example.com/v2/openstack/servers/
3cadb9e9-f430-4f62-8b9b-3efb671ff1fa" rel="self"/>
        <atom:link href="http://openstack.example.com/openstack/servers/3cadb9e9-
f430-4f62-8b9b-3efb671ff1fa" rel="bookmark"/>
    </server>
</servers>

```

This operation does not return a response body.

## 4.50.2. Show extended server attributes

Method	URI	Description
GET	/v2/servers/{server_id}	Shows extended server attributes for a specified server.

**Normal response codes:** 200

### 4.50.2.1. Request

This table shows the URI parameters for the show extended server attributes request:

Name	Type	Description
{server_id}	UUID	The UUID for the server of interest to you.

This operation does not accept a request body.

### 4.50.2.2. Response

#### Example 4.434. Show extended server attributes: JSON response

```
{
  "server": {
    "OS-EXT-SRV-ATTR:host": "1169a68456af48238da47b1d5957a714",
    "OS-EXT-SRV-ATTR:hypervisor_hostname": "fake-mini",
    "OS-EXT-SRV-ATTR:instance_name": "instance-00000001",
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": [
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    ],
    "created": "2012-11-15T19:27:04Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "2dfce43c41dd288cfac3a5b4251742b3bd2b37c12eb5927e757d9b4c",
    "id": "1fc2392e-5727-46af-bc21-317a4a3eb04c",
    "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/
1fc2392e-5727-46af-bc21-317a4a3eb04c",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/servers/
1fc2392e-5727-46af-bc21-317a4a3eb04c",
            "rel": "bookmark"
        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2012-11-15T19:27:04Z",
    "user_id": "fake"
}
}
}

```

### Example 4.435. Show extended server attributes: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-EXT-SRV-ATTR="http://docs.openstack.org/
compute/ext/extended_status/api/v1.1" xmlns:atom="http://www.
w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/
v1.1" status="ACTIVE" updated="2012-11-15T19:27:06Z" hostId=
"6b8205d183f40afad106dbeac44d3872151ef6f36790077ea2ea85fc" name="new-server-
test" created="2012-11-15T19:27:05Z" userId="fake" tenantId="openstack"
accessIPv4="" accessIPv6="" progress="0" id="ece641c1-51f5-4190-9342-
d9751f28eead" OS-EXT-SRV-ATTR:instance_name="instance-00000001" OS-
EXT-SRV-ATTR:host="80edfa5af48b4894b20eb1d9d2d4424e" OS-EXT-SRV-
ATTR:hypervisor_hostname="fake-mini">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" addr="192.168.0.3"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/
ece641c1-51f5-4190-9342-d9751f28eead" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/
ece641c1-51f5-4190-9342-d9751f28eead" rel="bookmark"/>

```

```
</server>
```

This operation does not return a response body.

## 4.51. Server extended status (servers)

Show extended status information, vm\_state, task\_state, and power\_state, in detailed server responses.

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/servers/{server_id}	Shows the extended status attributes in the response for a specified server.
<b>GET</b>	/v2/{tenant_id}/servers/detail	Lists the extended status attributes in the detailed response for all servers.

## 4.51.1. Show server extended status

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}	Shows the extended status attributes in the response for a specified server.

The extended status attributes are vm\_state, power\_state, and task\_state.

**Normal response codes:** 200

### 4.51.1.1. Request

This table shows the URI parameters for the show server extended status 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.

### 4.51.1.2. Response

#### Example 4.436. Show server extended status: JSON response

```
{
    "server": {
        "OS-EXT-STS:power_state": 1,
        "OS-EXT-STS:task_state": null,
        "OS-EXT-STS:vm_state": "active",
        "accessIPv4": "",
        "accessIPv6": "",
        "addresses": {
            "private": [
                {
                    "addr": "192.168.0.3",
                    "version": 4
                }
            ]
        },
        "created": "2013-02-07T19:35:09Z",
        "flavor": {
            "id": "1",
            "links": [
                {
                    "href": "http://openstack.example.com/openstack/flavors/1",
                    "rel": "bookmark"
                }
            ]
        },
        "hostId": "570eff4776ab310707d11d181037337197086998a8b3305c90bf87c8",
        "id": "ecb5e433-fa75-4db2-af3d-a29ae8618edc",
        "image": {
            "id": "70a599e0-31e7-49b7-b260-868f441e862b",
            "meta": [
                {
                    "key": "os-arch",
                    "value": "x86_64"
                },
                {
                    "key": "os-distro",
                    "value": "Ubuntu 12.04 LTS"
                }
            ],
            "minDisk": 1,
            "minMemory": 2048,
            "name": "Ubuntu 12.04 LTS"
        },
        "links": [
            {
                "href": "http://openstack.example.com/openstack/servers/ecb5e433-fa75-4db2-af3d-a29ae8618edc",
                "rel": "self"
            }
        ],
        "metadata": {
            "os-arch": "x86_64",
            "os-distro": "Ubuntu 12.04 LTS"
        },
        "name": "My Server"
    }
}
```

```

        "links": [
            {
                "href": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                "rel": "bookmark"
            }
        ],
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/servers/
ecb5e433-fa75-4db2-af3d-a29ae8618edc",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
ecb5e433-fa75-4db2-af3d-a29ae8618edc",
                "rel": "bookmark"
            }
        ],
        "metadata": {
            "My Server Name": "Apache1"
        },
        "name": "new-server-test",
        "progress": 0,
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2013-02-07T19:35:10Z",
        "user_id": "fake"
    }
}

```

### Example 4.437. Show server extended status: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-EXT-STS="http://docs.openstack.org/compute/
ext/extended_status/api/v1.1" xmlns:atom="http://www.w3.org/
2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.
1" status="ACTIVE" updated="2013-02-07T19:35:10Z" hostId=
"372afb648339fb6f22faa0b75fdd8834e2382fe02b352af8d7ee0b84" name=
"new-server-test" created="2013-02-07T19:35:09Z" userId="fake"
 tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id=
"68647408-85a7-4d9b-85e7-7f1e238983ad" OS-EXT-STS:vm_state="active" OS-EXT-
STS:task_state="None" OS-EXT-STS:power_state="1">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" addr="192.168.0.3"/>
        </network>
    </addresses>

```

```
<atom:link href="http://openstack.example.com/v2/openstack/servers/  
68647408-85a7-4d9b-85e7-7f1e238983ad" rel="self"/>  
<atom:link href="http://openstack.example.com/openstack/servers/  
68647408-85a7-4d9b-85e7-7f1e238983ad" rel="bookmark"/>  
</server>
```

## 4.51.2. List extended status for servers

Method	URI	Description
<b>GET</b>	/v2/{tenant_id}/servers/detail	Lists the extended status attributes in the detailed response for all servers.

The extended status attributes are `vm_state`, `power_state`, and `task_state`.

**Normal response codes:** 200

### 4.51.2.1. Request

This table shows the URI parameters for the list extended status for servers 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.

### 4.51.2.2. Response

#### Example 4.438. List extended status for servers: JSON response

```
{
    "servers": [
        {
            "OS-EXT-STS:power_state": 1,
            "OS-EXT-STS:task_state": null,
            "OS-EXT-STS:vm_state": "active",
            "accessIPv4": "",
            "accessIPv6": "",
            "addresses": [
                "private": [
                    {
                        "addr": "192.168.0.3",
                        "version": 4
                    }
                ]
            ],
            "created": "2012-12-05T07:34:10Z",
            "flavor": {
                "id": "1",
                "links": [
                    {
                        "href": "http://openstack.example.com/openstack/
flavors/1",
                        "rel": "bookmark"
                    }
                ]
            },
            "hostId": "585aa01f94eca692eff9f77ffe3eab866d8a819e97397e28c5c7df12",
            "id": "030758aa-5c41-41c6-8fb4-29d44eb96a85",
            "image": {
                "id": "70a599e0-31e7-49b7-b260-868f441e862b",
                "minDisk": 1,
                "minMemory": 2048,
                "name": "Ubuntu 12.04 LTS (Precise Pangolin) - CloudImage"
            }
        }
    ]
}
```

```

        "links": [
            {
                "href": "http://openstack.example.com/openstack/
images/70a599e0-31e7-49b7-b260-868f441e862b",
                "rel": "bookmark"
            }
        ],
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/
servers/030758aa-5c41-41c6-8fb4-29d44eb96a85",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
030758aa-5c41-41c6-8fb4-29d44eb96a85",
                "rel": "bookmark"
            }
        ],
        "metadata": {
            "My Server Name": "Apache1"
        },
        "name": "new-server-test",
        "progress": 0,
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2012-12-05T07:34:10Z",
        "user_id": "fake"
    }
]
}

```

### Example 4.439. List extended status for servers: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:OS-EXT-STS="http://docs.openstack.org/compute/ext/
extended_status/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2012-12-05T07:35:57Z" hostId=
"20171312b8f2c42b69b09360e08d7fe257b2e021107be687d0302a96" name="new-
server-test" created="2012-12-05T07:35:56Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="085c76aa-a58f-45b8-
ba78-4d1e541d5f89" OS-EXT-STS:vm_state="active" OS-EXT-STS:task_state="None"
OS-EXT-STS:power_state="1">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" addr="192.168.0.3" />
            </network>

```

```
</addresses>
<atom:link href="http://openstack.example.com/v2/openstack/servers/
085c76aa-a58f-45b8-ba78-4d1e541d5f89" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/servers/085c76aa-
a58f-45b8-ba78-4d1e541d5f89" rel="bookmark"/>
</server>
</servers>
```

## 4.52. Servers multiple create (servers)

Create one or more servers with an optional reservation ID. The request and response examples show how to create multiple servers with or without a reservation ID.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers{? return_reservation_id,min_count, max_count}	Creates one or more servers with an optional reservation ID.

## 4.52.1. Create multiple servers

Method	URI	Description
POST	/v2/{tenant_id}/servers{? return_reservation_id,min_count, max_count}	Creates one or more servers with an optional reservation ID.

**Normal response codes:** 202

### 4.52.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

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

Name	Type	Description
return_reservation_id	String (Optional)	Set to True to generate a reservation ID for each server. Omit this attribute to create servers without a reservation ID. This extended attribute is enabled when the service provider enables multiple server launch.
min_count	String (Optional)	The minimum number of servers to launch when the service provider enables multiple server launch.
max_count	String (Optional)	The maximum number of servers to launch when the service provider enables multiple server launch.

### Example 4.440. Create multiple servers: 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"
    },
    "min_count": "2",
    "max_count": "3",
    "personality": [
      {
        "path": "/etc/banner.txt",
        "contents": "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcyBpbBqdXN0IHN1Y2ggYSBkaXJ1Y3Rpb25"
      }
    ]
  }
}
```

{}

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

Name	Type	Description
security_groups  <i>(Optional)</i>	String	A security_groups object.
security_group  <i>(Optional)</i>	String	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.
user_data  <i>(Optional)</i>	String	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone  <i>(Optional)</i>	String	The availability zone in which to launch the server.
server  <i>(Required)</i>	ServerFor Create	server.
imageRef  <i>(Required)</i>	String	The image reference for the desired image for your server instance. Specify as an ID or full URL.
flavorRef  <i>(Required)</i>	String	The flavor reference for the desired flavor for your server instance. Specify as an ID or full URL.
networks  <i>(Optional)</i>	String	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant.  Optionally, you can create one or more NICs on the server.  To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a networks object.  To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a networks object.  You can specify multiple NICs on the server.
uuid  <i>(Optional)</i>	String	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a networks object. Required if you omit the <code>port</code> attribute.
port  <i>(Optional)</i>	String	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a networks object. Required if you omit the <code>uuid</code> attribute.
fixed_ip  <i>(Optional)</i>	String	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
name  <i>(Required)</i>	String	The server name.
metadata  <i>(Optional)</i>	String	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
personality  <i>(Optional)</i>	String	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
config_drive  <i>(Optional)</i>	String	Enables metadata injection in a server through a configuration drive.

Name	Type	Description
		To enable a configuration drive, specify true. Otherwise, specify false.

### Example 4.441. Create multiple servers: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
         imageRef="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b"
         flavorRef="http://openstack.example.com/openstack/flavors/1"
         name="new-server-test"
         min_count="2"
         max_count="3">
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <personality>
        <file path="/etc/banner.txt">
            ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
            dCBtb3ZlcYBpbIBqdXN0IHN1Y2ggYSBkaXJ1Y3Rpb24gYW5k
            IGF0IHN1Y2ggYSBzcGV1ZC4uLk10IGZ1ZWxzIGFuIGltcHVs
            c2lvbi4uLnRoaXMgaXMgdGh1IHByYWN1IHRvIGdvIG5vdy4g
            QnV0IHRoZSBza3kga25vd3MgdGh1IHJ1YXNvbnMgYW5kiHRo
            ZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3VkcYwgYW5kiHlV
            dSB3aWxsIGtub3csIHRvbywd2h1biB5b3UgbG1mdCB5b3Vy
            c2VsZiBoawdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
            b25zLiINCg0KLVJpY2hhcmQgQmFjaA==
        </file>
    </personality>
</server>
```

### Example 4.442. Create multiple servers: 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"
        },
        "return_reservation_id": "True",
        "min_count": "2",
        "max_count": "3",
        "personality": [
            {
                "path": "/etc/banner.txt",
                "contents":
                    "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcYBpbIBqdXN0IHN1Y2ggYSBkaXJ1Y3Rpb24gYW5kIHF0IHN1Y2ggYSBzcGV1ZC4uLk10IGZ1ZWxzIGFuIGltcHVs
                    c2lvbi4uLnRoaXMgaXMgdGh1IHByYWN1IHRvIGdvIG5vdy4g
                    QnV0IHRoZSBza3kga25vd3MgdGh1IHJ1YXNvbnMgYW5kiHRo
                    ZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3VkcYwgYW5kiHlV
                    dSB3aWxsIGtub3csIHRvbywd2h1biB5b3UgbG1mdCB5b3Vy
                    c2VsZiBoawdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
                    b25zLiINCg0KLVJpY2hhcmQgQmFjaA=="
            }
        ]
    }
}
```

### Example 4.443. Create multiple servers: XML request

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

```

<server xmlns="http://docs.openstack.org/compute/api/v1.1"
    imageRef="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b"
    flavorRef="http://openstack.example.com/openstack/flavors/1"
    name="new-server-test"
    min_count="2"
    max_count="3"
    return_reservation_id="True">
<metadata>
    <meta key="My Server Name">Apache1</meta>
</metadata>
<personality>
    <file path="/etc/banner.txt">
        ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
        dCBtb3ZlcBpbIBqdXN0IHN1Y2ggYSBkaXJ1Y3RpB24gYW5k
        IGF0IHN1Y2ggYSBzcgV1ZC4uLk10IGZ1ZWxzIGFuIGltcHVs
        c21vb14uLnRoaXMgaXMgdGh1IHBSYWN1IHRvIGdvIG5vdy4g
        QnV0IHRoZSBza3kga25vd3MgdGh1IHJ1YXNvbnMgYW5kIHRo
        ZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3VkcwYgYW5kIH1v
        dSB3aWxsIGtub3csIHRvbywgd2h1biB5b3UgbG1mdCB5b3Vy
        c2VsZiBoawdoIGVub3VnaCB0byBzZWUgYmV5b25kIGHvcml6
        b25zLiINCg0KLVJpY2hhcmQgQmFjaA==
    </file>
</personality>
</server>

```

This operation does not accept a request body.

#### 4.52.1.2. Response

##### Example 4.444. Create multiple servers: JSON response

```
{
    "server": {
        "adminPass": "wfksH3GTTsep",
        "id": "440cf918-3ee0-4143-b289-f63e1d2000e6",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/servers/
440cf918-3ee0-4143-b289-f63e1d2000e6",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
440cf918-3ee0-4143-b289-f63e1d2000e6",
                "rel": "bookmark"
            }
        ]
    }
}
```

This table shows the body parameters for the create multiple servers response:

Name	Type	Description
security_groups	String <i>(Optional)</i>	A security_groups object.
security_group	String <i>(Optional)</i>	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.

Name	Type	Description
user_data	String <i>(Optional)</i>	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String <i>(Optional)</i>	The availability zone in which to launch the server.
server	ServerFor Create <i>(Required)</i>	server.
imageRef	String <i>(Required)</i>	The image reference for the desired image for your server instance. Specify as an ID or full URL.
flavorRef	String <i>(Required)</i>	The flavor reference for the desired flavor for your server instance. Specify as an ID or full URL.
networks	String <i>(Optional)</i>	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant.  Optionally, you can create one or more NICs on the server.  To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a networks object.  To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a networks object.  You can specify multiple NICs on the server.
uuid	String <i>(Optional)</i>	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a networks object. Required if you omit the <code>port</code> attribute.
port	String <i>(Optional)</i>	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a networks object. Required if you omit the <code>uuid</code> attribute.
fixed_ip	String <i>(Optional)</i>	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
name	String <i>(Required)</i>	The server name.
metadata	String <i>(Optional)</i>	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
personality	String <i>(Optional)</i>	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
config_drive	String <i>(Optional)</i>	Enables metadata injection in a server through a configuration drive.  To enable a configuration drive, specify <code>true</code> . Otherwise, specify <code>false</code> .

### Example 4.445. Create multiple servers: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" id="0857bea2-df83-4810-876a-093a6b9afb15" adminPass="edbBzA3AnJP67">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/0857bea2-df83-4810-876a-093a6b9afb15" rel="self"/>
```

```
<atom:link href="http://openstack.example.com/openstack/servers/0857bea2-
df83-4810-876a-093a6b9afb15" rel="bookmark"/>
</server>
```

#### Example 4.446. Create multiple servers: JSON response

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

#### Example 4.447. Create multiple servers: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1" reservation_id="r-fq0lg4za"/>
```

This operation does not return a response body.

## 4.53. Server rescue and unrescue (servers)

Put a server into rescue mode or unrescue a server in rescue mode.

Method	URI	Description
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Puts a server in rescue mode. Changes status to RESCUE.
<b>POST</b>	/v2/{tenant_id}/servers/{server_id}/action	Unrescues a server.

## 4.53.1. Rescue server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Puts a server in rescue mode. Changes status to RESCUE.

Normal response codes: 200

### 4.53.1.1. Request

This table shows the URI parameters for the rescue server 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 4.448. Rescue server: JSON request

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

This table shows the body parameters for the rescue server request:

Name	Type	Description
rescue	String <i>(Required)</i>	Specify the rescue action in the request body.
adminPass	String <i>(Optional)</i>	Use this password for the rescued instance. Generate a new password if none is provided.

#### Example 4.449. Rescue server: XML request

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

This operation does not accept a request body.

### 4.53.1.2. Response

#### Example 4.450. Rescue server: JSON response

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

#### Example 4.451. Rescue server: XML response

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

This operation does not return a response body.

## 4.53.2. Unrescue server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Unrescues a server.

Normal response codes: 202

### 4.53.2.1. Request

This table shows the URI parameters for the unrescue server 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 4.452. Unrescue server: JSON request

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

This table shows the body parameters for the unrescue server request:

Name	Type	Description
unrescue	String <i>(Required)</i>	Specify the unrescue action in the request body.

#### Example 4.453. Unrescue server: XML request

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

This operation does not accept a request body.

## 4.54. Servers with scheduler hints (servers)

Create a server with scheduler hints.

Method	URI	Description
POST	/v2/{tenant_id}/servers	Creates a server with scheduler hints that are passed directly to the scheduler.

## 4.54.1. Create server with scheduler hints

Method	URI	Description
POST	/v2/{tenant_id}/servers	Creates a server with scheduler hints that are passed directly to the scheduler.

Normal response codes: 200

### 4.54.1.1. Request

This table shows the URI parameters for the create server with scheduler hints request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

#### Example 4.454. Create server with scheduler hints: JSON request

```
{
  "server": {
    "name": "new-server-test",
    "imageRef": "70a599e0-31e7-49b7-b260-868f441e862b",
    "flavorRef": "1"
  },
  "os:scheduler_hints": {
    "hypervisor": "xen",
    "near": "2b7c42eb-7736-4a0f-afab-f23969a35ada"
  }
}
```

This table shows the body parameters for the create server with scheduler hints request:

Name	Type	Description
security_groups <i>(Optional)</i>	String	A security_groups object.
security_group <i>(Optional)</i>	String	One or more security_group objects. Specify the name of the security group in the name attribute. If you omit this attribute, the server is created in the default security group.
user_data <i>(Optional)</i>	String	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone <i>(Optional)</i>	String	The availability zone in which to launch the server.
server <i>(Required)</i>	ServerForCreate	server.
imageRef <i>(Required)</i>	String	The image reference for the desired image for your server instance. Specify as an ID or full URL.
flavorRef <i>(Required)</i>	String	The flavor reference for the desired flavor for your server instance. Specify as an ID or full URL.
networks	String	A networks object. By default, the server instance is provisioned with all isolated networks for the tenant.

Name	Type	Description
	(Optional)	<p>Optionally, you can create one or more NICs on the server.</p> <p>To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object.</p> <p>To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object.</p> <p>You can specify multiple NICs on the server.</p>
<code>uuid</code>	String (Optional)	To provision the server instance with a NIC for a nova-network network, specify the UUID in the <code>uuid</code> attribute in a <code>networks</code> object. Required if you omit the <code>port</code> attribute.
<code>port</code>	String (Optional)	To provision the server instance with a NIC for a neutron network, specify the UUID in the <code>port</code> attribute in a <code>networks</code> object. Required if you omit the <code>uuid</code> attribute.
<code>fixed_ip</code>	String (Optional)	A fixed IPv4 address for the NIC. Valid with a neutron or nova-networks network.
<code>name</code>	String (Required)	The server name.
<code>metadata</code>	String (Optional)	Metadata key and value pairs. The maximum size of the metadata key and value is 255 bytes each.
<code>personality</code>	String (Optional)	File path and contents (text only) to inject into the server at launch. The maximum size of the file path data is 255 bytes. The maximum limit refers to the number of bytes in the decoded data and not the number of characters in the encoded data.
<code>config_drive</code>	String (Optional)	Enables metadata injection in a server through a configuration drive. To enable a configuration drive, specify <code>true</code> . Otherwise, specify <code>false</code> .
<code>os:scheduler_hints</code>	Dict (Required)	Dictionary of data to send to the scheduler.

### Example 4.455. Create server with scheduler hints: XML request

```

<server
    xmlns="http://docs.openstack.org/compute/api/v1.1"
    xmlns:OS-SCH-HNT="http://docs.openstack.org/compute/ext/scheduler-
hints/api/v2"
        name='new-server-test'
        imageRef='70a599e0-31e7-49b7-b260-868f441e862b'
        flavorRef='1'
    >
    <OS-SCH-HNT:scheduler_hints>
        <hypervisor>xen</hypervisor>
        <near>eb999657-dd6b-464e-8713-95c532ac3b18</near>
    </OS-SCH-HNT:scheduler_hints>
</server>
```

This operation does not accept a request body.

#### 4.54.1.2. Response

### Example 4.456. Create server with scheduler hints: JSON response

```
{
```

```
"server": {
    "adminPass": "yjzytFHb7XHc",
    "id": "f8f4f3ce-f6e0-4e05-8f79-bf984fdfce45",
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/servers/f8f4f3ce-f6e0-4e05-8f79-bf984fdfce45",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/servers/f8f4f3ce-f6e0-4e05-8f79-bf984fdfce45",
            "rel": "bookmark"
        }
    ]
}
```

#### Example 4.457. Create server with scheduler hints: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" id="a14a9bf5-93b3-49e4-85f1-01d7cf2645b4" adminPass="Q6jfHPrnvB9X">
    <metadata/>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/a14a9bf5-93b3-49e4-85f1-01d7cf2645b4" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/a14a9bf5-93b3-49e4-85f1-01d7cf2645b4" rel="bookmark"/>
</server>
```

This operation does not return a response body.

# 5. Compute API v3 (EXPERIMENTAL)



## Note

XML support in requests and responses has been deprecated for the Compute API v3.

Method	URI	Description
API versions		
<b>GET</b>	/	Lists information about all Compute API versions.
<b>GET</b>	/v3	Shows details for Compute API v3.
Server admin actions (servers)		
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Backs up a server instance.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Injects network information into a server.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Live-migrates a server to a new host without rebooting.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Lock a server instance.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Migrates a server to a host. The scheduler chooses the host.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Pauses a server. Changes its status to PAUSED.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Resets networking on a server.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Resumes a SUSPENDED server and changes its status to ACTIVE.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Suspends a server and changes its status to SUSPENDED.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Unlocks a server instance.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Unpauses a PAUSED server and changes its status to ACTIVE.
Guest agents (os-agents)		
<b>POST</b>	/v3/os-agents	Creates an agent build.
<b>GET</b>	/v3/os-agents	Lists agent builds.
<b>PUT</b>	/v3/os-agents/{id}	Updates an agent build.
Host aggregates (os-aggregates)		
<b>POST</b>	/v3/os-aggregates	Creates an aggregate, given its name and availability zone.
<b>GET</b>	/v3/os-aggregates	Lists aggregates id, name, and availability_zone for an aggregate.
<b>GET</b>	/v3/os-aggregates/{aggregate_id}	Shows the details of an aggregate, hosts and metadata included.
<b>GET</b>	/v3/os-aggregates/{aggregate_id}	Updates either or both the name and availability zone for a specified aggregate.
<b>POST</b>	/v3/os-aggregates/{aggregate_id}/action	Adds a host to the specified aggregate.
<b>POST</b>	/v3/os-aggregates/{aggregate_id}/action	Creates or replaces metadata for an aggregate.

Method	URI	Description
<b>POST</b>	/v3/os-aggregates/{aggregate_id}/action	Removes a host from the specified aggregate.
Cells (os-cells)		
<b>GET</b>	/v3/os-cells	Lists cells.
<b>GET</b>	/v3/os-cells	Lists cells with details.
<b>GET</b>	/v3/os-cells/cell3	Shows data for a specified cell.
<b>GET</b>	/v3/os-cells/nova/capacities	Shows capacities for a specified cell.
Root certificates (os-certificates)		
<b>POST</b>	/v3/os-certificates	Creates a certificate.
<b>GET</b>	/v3/os-certificates/root	Shows details for a specified certificate.
Configuration drive (os-config-drive)		
<b>GET</b>	/v3/servers/{server_id}	Shows details for a specified server.
<b>GET</b>	/v3/servers/detail	Lists server details for a specified user.
Server deferred delete (os-deferred-delete)		
<b>POST</b>	/v3/servers/{server_id}/action	Force deletes an instance before deferred cleanup.
<b>POST</b>	/v3/servers/{server_id}/action	Restores a previously deleted instance.
Evacuate (os-evacuate)		
<b>POST</b>	/v3/servers/{server_id}/action	Evacuates a server from a failed host to a new one.
Servers with extended availability zones (os-extended-availability-zone)		
<b>GET</b>	/v3/servers/{server_id}	Shows details for a specified server.
<b>GET</b>	/v3/servers/detail	Lists server details for a specified user.
Server extended attributes (os-extended-server-attributes)		
<b>GET</b>	/v3/servers	Shows details for a specified server.
<b>GET</b>	/v3/servers/details	Lists server details for a specified user.
Server extended status (os-extended-status)		
<b>GET</b>	/v3/servers/{server_id}	Shows details for a specified server.
<b>GET</b>	/v3/servers/detail	Lists servers with details for a specified user.
Flavor access (os-flavor-access)		
<b>POST</b>	/v3/flavors	Adds access attribute to the flavor create response.
<b>GET</b>	/v3/flavors/detail	Extends flavor detail to add access attribute to the response of flavor detail.
<b>GET</b>	/v3/flavors/1	Extends flavor show to add access attribute to the flavor show response.
<b>POST</b>	/v3/flavors/10/action	Adds flavor access for tenant.
<b>POST</b>	/v3/flavors/10/action	Removes flavor access for tenant.
<b>GET</b>	/v3/flavors/10/os-flavor-access	Returns access list by flavor id.
Flavor extra-specs (flavor-extra-specs)		
<b>POST</b>	/v3/flavors/{flavor_id}/flavor-extra-specs	Creates and updates flavor extra specs.
<b>GET</b>	/v3/flavors/{flavor_id}/flavor-extra-specs	Lists extra specs for specified flavor.
<b>GET</b>	/v3/flavors/{flavor_id}/flavor-extra-specs/{flavor_extra_spec_key}	Shows an extra spec for specified flavor by the key.
<b>PUT</b>	/v3/flavors/{flavor_id}/flavor-extra-specs/{flavor_extra_spec_key}	Updates specified extra spec value by the key.
Flavors manage (flavor-manage)		
<b>POST</b>	/v3/flavors	Creates a flavor.

Method	URI	Description
<b>DELETE</b>	/v3/flavors/{flavor_id}	Deletes a flavor.
Flavors with rxtx_factor extended attribute (os-flavor-rxtx)		
<b>POST</b>	/v3/flavors	Creates a flavor.
<b>GET</b>	/v3/flavors/detail	Lists flavors with details.
<b>GET</b>	/v3/flavors/{flavor_id}	Shows details for a specified flavor.
Flavors with extended attributes (flavors)		
<b>GET</b>	/v3/flavors	Lists flavors.
<b>GET</b>	/v3/flavors/{flavor_id}	Shows information for a specified flavor.
<b>GET</b>	/v3/flavors/detail	Lists flavors with details.
Hosts (os-hosts)		
<b>GET</b>	/v3/os-hosts	Lists hosts.
<b>PUT</b>	/v3/os-hosts/{host_name}	Enables or puts a host in maintenance mode.
<b>GET</b>	/v3/os-hosts/{host_name}	Shows details for a specified host.
<b>GET</b>	/v3/os-hosts/{host_name}/reboot	Reboots a host.
<b>GET</b>	/v3/os-hosts/{host_name}/shutdown	Shuts down a host.
<b>GET</b>	/v3/os-hosts/{host_name}/startup	Starts a host.
Hypervisors (os-hypervisors)		
<b>GET</b>	/v3/os-hypervisors	Lists hypervisors.
<b>GET</b>	/v3/os-hypervisors/statistics	Shows statistics for all hypervisors.
<b>GET</b>	/v3/os-hypervisors/search{?query}	Searches hypervisors by specified host name.
<b>GET</b>	/v3/os-hypervisors/{hypervisor_id}	Shows details for a specified hypervisor.
<b>GET</b>	/v3/os-hypervisors/{hypervisor_id}/uptime	Shows the uptime for a specified hypervisor.
<b>GET</b>	/v3/os-hypervisors/{hypervisor_id}/servers	Lists servers that run on a specified hypervisor.
Server actions (servers)		
<b>GET</b>	/v3/servers/{server_id}/os-instance-actions	Lists actions for a specified instance.
<b>GET</b>	/v3/servers/{server_id}/os-instance-actions/{request_id}	Shows information about a specified instance action.
Instance usage audit log (os-instance-usage-audit-log)		
<b>GET</b>	/v3/os-instance-usage-audit-log	Lists usage audits for a specified instance.
<b>GET</b>	/v3/os-instance-usage-audit-log/{datetime}	Lists instance usage audits that occurred before a specified time.
Limits (limits)		
<b>GET</b>	/v3/limits	Show global and rate limit information.
Migrations (os-migrations)		
<b>GET</b>	/v3/os-migrations	Lists in-progress migrations.
Multinic (os-multinic)		
<b>POST</b>	/v3/servers	Removes an IP from a specified instance.
<b>POST</b>	/v3/servers/action	Adds an IP to a specified network on an instance.
Quota class (os-quota-class-sets)		
<b>GET</b>	/v3/os-quota-class-sets/{class_id}	Shows the quota for a specified class.
<b>PUT</b>	/v3/os-quota-class-sets/{class_id}	Updates quota for a specified class.
Quota sets (os-quota-sets)		
<b>DELETE</b>	/v3/os-quota-sets/{tenant_id}	Deletes a quota for tenant.

Method	URI	Description
<b>GET</b>	/v3/os-quota-sets/{tenant_id}	Shows quotas for tenant.
<b>PUT</b>	/v3/os-quota-sets/{tenant_id}	Force-updates quota for tenant.
<b>PUT</b>	/v3/os-quota-sets/{tenant_id}	Updates quota for tenant.
<b>GET</b>	/v3/os-quota-sets/{tenant_id}/defaults	Shows default quotas for tenant.
<b>PUT</b>	/v3/os-quota-sets/{tenant_id}/{?user_id}	Updates quota for user.
<b>DELETE</b>	/v3/os-quota-sets/{tenant_id}/{?user_id}	Deletes quota for a specified user.
Server remote console (os-remote-consoles)		
<b>POST</b>	/v3/servers/{server_id}/actions	Gets text console output.
<b>POST</b>	/v3/servers/{server_id}/actions	Gets text console output for VNC.
Server usage (os-server-usage)		
<b>GET</b>	/v3/servers/detail	Lists server details for a specified user.
<b>GET</b>	/v3/servers/{server_id}	Shows details for a specified server.
Usage reports (os-simple-tenant-usage)		
<b>GET</b>	/v3/os-simple-tenant-usage	Lists usage information for all tenants.
<b>GET</b>	/v3/os-simple-tenant-usage/{tenant_id}	Shows usage details for a specified tenant.

## 5.1. API versions

List information for all API versions.

Method	URI	Description
<b>GET</b>	/	Lists information about all Compute API versions.
<b>GET</b>	/v3	Shows details for Compute API v3.

## 5.1.1. List API versions

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

**Normal response codes:** 200, 300

### 5.1.1.1. Request

This operation does not accept a request body.

### 5.1.1.2. Response

#### Example 5.1. List API versions: JSON response

```
{
  "versions": [
    {
      "status": "CURRENT",
      "updated": "2011-01-21T11:33:21Z",
      "id": "v2.0",
      "links": [
        {
          "href": "http://23.253.228.211:8774/v2/",
          "rel": "self"
        }
      ]
    },
    {
      "status": "EXPERIMENTAL",
      "updated": "2013-07-23T11:33:21Z",
      "id": "v3.0",
      "links": [
        {
          "href": "http://23.253.228.211:8774/v3/",
          "rel": "self"
        }
      ]
    }
  ]
}
```

#### Example 5.2. List API versions: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<choices xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/common/api/v1.0">
  <version status="EXPERIMENTAL" id="v3.0">
    <media-types>
      <media-type base="application/json"
        type="application/vnd.openstack.compute+json;version=3"
      />
    </media-types>
    <atom:link href="http://23.253.228.211:8774/v3/.xml"
      rel="self"/>
  </version>
```

```
<version status="CURRENT" id="v2.0">
  <media-types>
    <media-type base="application/xml"
      type="application/vnd.openstack.compute+xml;version=2"/>
    <media-type base="application/json"
      type="application/vnd.openstack.compute+json;version=2"
    />
  </media-types>
  <atom:link href="http://23.253.228.211:8774/v2/.xml"
    rel="self"/>
</version>
</choices>
```

This operation does not return a response body.

## 5.1.2. Show API v3 details

Method	URI	Description
GET	/v3	Shows details for Compute API v3.

**Normal response codes:** 200, 203

### 5.1.2.1. Request

This operation does not accept a request body.

### 5.1.2.2. Response

#### Example 5.3. Show API v3 details: JSON response

```
{
  "version": {
    "status": "EXPERIMENTAL",
    "updated": "2013-07-23T11:33:21Z",
    "media-types": [
      {
        "base": "application/json",
        "type": "application/vnd.openstack.compute+json;version=3"
      }
    ],
    "id": "v3.0",
    "links": [
      {
        "href": "http://23.253.228.211:8774/v3/",
        "rel": "self"
      },
      {
        "href": "http://docs.openstack.org/api/openstack-compute/3/os-compute-devguide-3.pdf",
        "type": "application/pdf",
        "rel": "describedby"
      },
      {
        "href": "http://docs.openstack.org/api/openstack-compute/3/wadl/os-compute-3.wadl",
        "type": "application/vnd.sun.wadl+xml",
        "rel": "describedby"
      }
    ]
  }
}
```

This table shows the body parameters for the show api v3 details response:

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

#### Example 5.4. Show API v3 details: XML response

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

```

<version>
  <status>EXPERIMENTAL</status>
  <updated>2013-07-23T11:33:21Z</updated>
  <media-types>
    <media-type>
      <base>application/json</base>
      <type>application/vnd.openstack.compute+json;version=3</type>
    </media-type>
  </media-types>
  <id>v3.0</id>
  <links>
    <link>
      <href>http://23.253.228.211:8774/v3/</href>
      <rel>self</rel>
    </link>
    <link>
      <href>http://docs.openstack.org/api/openstack-compute/3/os-compute-devguide-3.pdf</href>
      <type>application/pdf</type>
      <rel>describedby</rel>
    </link>
    <link>
      <href>http://docs.openstack.org/api/openstack-compute/3/wadl/os-compute-3.wadl</href>
      <type>application/vnd.sun.wadl+xml</type>
      <rel>describedby</rel>
    </link>
  </links>
</version>

```

This operation does not return a response body.

## 5.2. Server admin actions (servers)

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

Method	URI	Description
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Backs up a server instance.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Injects network information into a server.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Live-migrates a server to a new host without rebooting.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Lock a server instance.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Migrates a server to a host. The scheduler chooses the host.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Pauses a server. Changes its status to PAUSED.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Resets networking on a server.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Resumes a SUSPENDED server and changes its status to ACTIVE.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Suspends a server and changes its status to SUSPENDED.
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Unlocks a server instance.

Method	URI	Description
<b>POST</b>	/v3/servers/{server_id}/action/{server_id}/action	Unpauses a PAUSED server and changes its status to ACTIVE.

## 5.2.1. Back up server

Method	URI	Description
POST	/v3/servers/{server_id}/action/{server_id}/action	Backs up a server instance.

Normal response codes: 202

### 5.2.1.1. Request

#### Example 5.5. Back up server: JSON request

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

#### Example 5.6. Back up server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>  
  <create_backup>  
    <name>Backup 1</name>  
    <backup_type>daily</backup_type>  
    <rotation>1</rotation>  
  </create_backup>
```

This operation does not accept a request body.

## 5.2.2. Inject network information

Method	URI	Description
POST	/v3/servers/{server_id}/action/{server_id}/action	Injects network information into a server.

**Normal response codes:** 202

### 5.2.2.1. Request

#### Example 5.7. Inject network information: JSON request

```
{  
    "inject_network_info": null  
}
```

#### Example 5.8. Inject network information: XML request

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

This operation does not accept a request body.

## 5.2.3. Live migrate to new host

Method	URI	Description
POST	/v3/servers/{server_id}/action/{server_id}/action	Live-migrates a server to a new host without rebooting.

Normal response codes: 202

### 5.2.3.1. Request

#### Example 5.9. Live migrate to new host: JSON request

```
{  
    "migrate_live": {  
        "host": "01c0cadef72d47e28a672a76060d492c",  
        "block_migration": false,  
        "disk_over_commit": false  
    }  
}
```

#### Example 5.10. Live migrate to new host: XML request

```
<?xml version="1.0" encoding="UTF-8" ?>  
  <migrate_live>  
    <host>bba598d96c9a465e80b1455f5fa8d364</host>  
    <block_migration>false</block_migration>  
    <disk_over_commit>false</disk_over_commit>  
  </migrate_live>
```

This operation does not accept a request body.

## 5.2.4. Lock server

Method	URI	Description
POST	/v3/servers/{server_id}/action/{server_id}/action	Lock a server instance.

Normal response codes: 202

### 5.2.4.1. Request

#### Example 5.11. Lock server: JSON request

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

#### Example 5.12. Lock server: XML request

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

This operation does not accept a request body.

## 5.2.5. Migrate server

Method	URI	Description
POST	/v3/servers/{server_id}/action/{server_id}/action	Migrates a server to a host. The scheduler chooses the host.

Normal response codes: 202

### 5.2.5.1. Request

#### Example 5.13. Migrate server: JSON request

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

#### Example 5.14. Migrate server: XML request

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

This operation does not accept a request body.

## 5.2.6. Pause server

Method	URI	Description
POST	/v3/servers/{server_id}/action/{server_id}/action	Pauses a server. Changes its status to PAUSED.

**Normal response codes:** 202

### 5.2.6.1. Request

#### Example 5.15. Pause server: JSON request

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

#### Example 5.16. Pause server: XML request

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

This operation does not accept a request body.

## 5.2.7. Reset server networking

Method	URI	Description
POST	/v3/servers/{server_id}/action/{server_id}/action	Resets networking on a server.

**Normal response codes:** 202

### 5.2.7.1. Request

#### Example 5.17. Reset server networking: JSON request

```
{  
    "reset_network": null  
}
```

#### Example 5.18. Reset server networking: XML request

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

This operation does not accept a request body.

## 5.2.8. Resume server

Method	URI	Description
POST	/v3/servers/{server_id}/action/{server_id}/action	Resumes a SUSPENDED server and changes its status to ACTIVE.

**Normal response codes:** 202

### 5.2.8.1. Request

#### Example 5.19. Resume server: JSON request

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

#### Example 5.20. Resume server: XML request

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

This operation does not accept a request body.

## 5.2.9. Suspend server

Method	URI	Description
POST	/v3/servers/{server_id}/action/{server_id}/action	Suspends a server and changes its status to SUSPENDED.

**Normal response codes:** 202

### 5.2.9.1. Request

#### Example 5.21. Suspend server: JSON request

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

#### Example 5.22. Suspend server: XML request

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

This operation does not accept a request body.

## 5.2.10. Unlock server

Method	URI	Description
POST	/v3/servers/{server_id}/action/{server_id}/action	Unlocks a server instance.

**Normal response codes:** 202

### 5.2.10.1. Request

#### Example 5.23. Unlock server: JSON request

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

#### Example 5.24. Unlock server: XML request

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

This operation does not accept a request body.

## 5.2.11. Unpause server

Method	URI	Description
POST	/v3/servers/{server_id}/action/{server_id}/action	Unpauses a PAUSED server and changes its status to ACTIVE.

Normal response codes: 202

### 5.2.11.1. Request

#### Example 5.25. Unpause server: JSON request

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

#### Example 5.26. Unpause server: XML request

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

This operation does not accept a request body.

## 5.3. Guest agents (os-agents)

Creates, updates, and deletes guest agents. Use guest agents to access files on the disk, configure networking, or run other applications or scripts in the guest while it 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	/v3/os-agents	Creates an agent build.
GET	/v3/os-agents	Lists agent builds.
PUT	/v3/os-agents/{id}	Updates an agent build.

## 5.3.1. Create agent build

Method	URI	Description
POST	/v3/os-agents	Creates an agent build.

**Normal response codes:** 201

### 5.3.1.1. Request

#### Example 5.27. Create agent build: JSON request

```
{
  "agent": {
    "hypervisor": "hypervisor",
    "os": "os",
    "architecture": "x86",
    "version": "8.0",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "url": "xxxxxxxxxxxxxx"
  }
}
```

#### Example 5.28. Create agent build: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<agent>
  <hypervisor>hypervisor</hypervisor>
  <os>os</os>
  <architecture>x86</architecture>
  <version>8.0</version>
  <md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
  <url>xxxxxxxxxxxxxx</url>
</agent>
```

This operation does not accept a request body.

### 5.3.1.2. Response

#### Example 5.29. Create agent build: JSON response

```
{
  "agent": {
    "agent_id": "1",
    "architecture": "x86",
    "hypervisor": "hypervisor",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "os": "os",
    "url": "xxxxxxxxxxxxxx",
    "version": "8.0"
  }
}
```

#### Example 5.30. Create agent build: XML response

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

```
<agent>
  <url>xxxxxxxxxxxx</url>
  <hypervisor>hypervisor</hypervisor>
  <md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
  <version>8.0</version>
  <architecture>x86</architecture>
  <os>os</os>
  <agent_id>1</agent_id>
</agent>
```

This operation does not return a response body.

## 5.3.2. List agent builds

Method	URI	Description
GET	/v3/os-agents	Lists agent builds.

**Normal response codes:** 200

### 5.3.2.1. Request

This operation does not accept a request body.

### 5.3.2.2. Response

#### Example 5.31. List agent builds: JSON response

```
{  
    "agents": [  
        {  
            "agent_id": "1",  
            "architecture": "x86",  
            "hypervisor": "hypervisor",  
            "md5hash": "add6bb58e139be103324d04d82d8f545",  
            "os": "os",  
            "url": "xxxxxxxxxxxxxx",  
            "version": "8.0"  
        }  
    ]  
}
```

#### Example 5.32. List agent builds: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<agents>  
    <agent url="xxxxxxxxxxxxxx" hypervisor="hypervisor" md5hash=  
"add6bb58e139be103324d04d82d8f545" version="8.0" architecture="x86" os="os"  
    agent_id="1"/>  
</agents>
```

This operation does not return a response body.

### 5.3.3. Update agent build

Method	URI	Description
PUT	/v3/os-agents/{id}	Updates an agent build.

**Normal response codes:** 200

#### 5.3.3.1. Request

##### Example 5.33. Update agent build: JSON request

```
{
  "agent": {
    "url": "xxx://xxxx/xxx/xxx",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "version": "7.0"
  }
}
```

##### Example 5.34. Update agent build: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<agent>
  <version>7.0</version>
  <url>xxx://xxxx/xxx/xxx</url>
  <md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
</agent>
```

This operation does not accept a request body.

#### 5.3.3.2. Response

##### Example 5.35. Update agent build: JSON response

```
{
  "agent": {
    "agent_id": "1",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "url": "xxx://xxxx/xxx/xxx",
    "version": "7.0"
  }
}
```

##### Example 5.36. Update agent build: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<agent>
  <url>xxx://xxxx/xxx/xxx</url>
  <version>7.0</version>
  <agent_id>1</agent_id>
  <md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
</agent>
```

This operation does not return a response body.

## 5.4. Host aggregates (os-aggregates)

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
<b>POST</b>	/v3/os-aggregates	Creates an aggregate, given its name and availability zone.
<b>GET</b>	/v3/os-aggregates	Lists aggregates id, name, and availability_zone for an aggregate.
<b>GET</b>	/v3/os-aggregates/{aggregate_id}	Shows the details of an aggregate, hosts and metadata included.
<b>GET</b>	/v3/os-aggregates/{aggregate_id}	Updates either or both the name and availability zone for a specified aggregate.
<b>POST</b>	/v3/os-aggregates/{aggregate_id}/action	Adds a host to the specified aggregate.
<b>POST</b>	/v3/os-aggregates/{aggregate_id}/action	Creates or replaces metadata for an aggregate.
<b>POST</b>	/v3/os-aggregates/{aggregate_id}/action	Removes a host from the specified aggregate.

## 5.4.1. Create aggregate

Method	URI	Description
POST	/v3/os-aggregates	Creates an aggregate, given its name and availability zone.

**Normal response codes:** 201

### 5.4.1.1. Request

#### Example 5.37. Create aggregate: JSON request

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

#### Example 5.38. Create aggregate: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<aggregate name="name" availability_zone="nova" />
```

This operation does not accept a request body.

### 5.4.1.2. Response

#### Example 5.39. Create aggregate: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2013-08-18T12:17:55.751757",
    "deleted": 0,
    "deleted_at": null,
    "id": 1,
    "name": "name",
    "updated_at": null
  }
}
```

#### Example 5.40. Create aggregate: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
  <name>name</name>
  <availability_zone>nova</availability_zone>
  <deleted>False</deleted>
  <created_at>2013-08-18 12:17:56.757058</created_at>
  <updated_at>None</updated_at>
  <deleted_at>None</deleted_at>
  <id>1</id>
</aggregate>
```

This operation does not return a response body.

## 5.4.2. List aggregates

Method	URI	Description
GET	/v3/os-aggregates	Lists aggregates id, name, and availability_zone for an aggregate.

Normal response codes: 200

### 5.4.2.1. Request

#### Example 5.41. List aggregates: JSON request

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

#### Example 5.42. List aggregates: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<aggregate name="name" availability_zone="nova" />
```

This operation does not accept a request body.

### 5.4.2.2. Response

#### Example 5.43. List aggregates: JSON response

```
{
  "aggregates": [
    {
      "availability_zone": "nova",
      "created_at": "2013-08-18T12:17:56.856455",
      "deleted": 0,
      "deleted_at": null,
      "hosts": [],
      "id": 1,
      "metadata": {
        "availability_zone": "nova"
      },
      "name": "name",
      "updated_at": null
    }
  ]
}
```

#### Example 5.44. List aggregates: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregates>
  <aggregate>
    <name>name</name>
    <availability_zone>nova</availability_zone>
```

```
<deleted>False</deleted>
<created_at>2013-08-18 12:17:57.241412</created_at>
<updated_at>None</updated_at>
<hosts/>
<deleted_at>None</deleted_at>
<id>1</id>
<metadata>
    <availability_zone>nova</availability_zone>
</metadata>
</aggregate>
</aggregates>
```

This operation does not return a response body.

## 5.4.3. Show aggregate details

Method	URI	Description
GET	/v3/os-aggregates/{aggregate_id}	Shows the details of an aggregate, hosts and metadata included.

**Normal response codes:** 200

### 5.4.3.1. Request

#### Example 5.45. Show aggregate details: JSON request

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

#### Example 5.46. Show aggregate details: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<aggregate name="name" availability_zone="nova" />
```

This operation does not accept a request body.

### 5.4.3.2. Response

#### Example 5.47. Show aggregate details: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2013-08-18T12:17:56.380226",
    "deleted": 0,
    "deleted_at": null,
    "hosts": [],
    "id": 1,
    "metadata": {
      "availability_zone": "nova"
    },
    "name": "name",
    "updated_at": null
  }
}
```

#### Example 5.48. Show aggregate details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
  <name>name</name>
  <availability_zone>nova</availability_zone>
  <deleted>False</deleted>
  <created_at>2013-08-18 12:17:56.757058</created_at>
  <updated_at>None</updated_at>
```

```
<hosts/>
<deleted_at>None</deleted_at>
<id>1</id>
<metadata>
  <availability_zone>nova</availability_zone>
</metadata>
</aggregate>
```

This operation does not return a response body.

## 5.4.4. Update aggregate

Method	URI	Description
GET	/v3/os-aggregates/{aggregate_id}	Updates either or both the name and availability zone for a specified aggregate.

**Normal response codes:** 200

### 5.4.4.1. Request

#### Example 5.49. Update aggregate: JSON request

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

#### Example 5.50. Update aggregate: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<aggregate name="newname" availability_zone="nova2" />
```

This operation does not accept a request body.

### 5.4.4.2. Response

#### Example 5.51. Update aggregate: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova2",
    "created_at": "2013-08-18T12:17:56.259751",
    "deleted": 0,
    "deleted_at": null,
    "hosts": [],
    "id": 1,
    "metadata": {
      "availability_zone": "nova2"
    },
    "name": "newname",
    "updated_at": "2013-08-18T12:17:56.286720"
  }
}
```

#### Example 5.52. Update aggregate: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
  <name>newname</name>
  <availability_zone>nova2</availability_zone>
  <deleted>False</deleted>
  <created_at>2013-08-18 12:17:57.655769</created_at>
  <updated_at>2013-08-18 12:17:57.683956</updated_at>
```

```
<hosts/>
<deleted_at>None</deleted_at>
<id>1</id>
<metadata>
  <availability_zone>nova2</availability_zone>
</metadata>
</aggregate>
```

This operation does not return a response body.

## 5.4.5. Add host

Method	URI	Description
POST	/v3/os-aggregates/{aggregate_id}/action	Adds a host to the specified aggregate.

**Normal response codes:** 202

### 5.4.5.1. Request

#### Example 5.53. Add host: JSON request

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

#### Example 5.54. Add host: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<add_host host="bdb794e5cdd3447c9132ddd8e5cc3d0b" />
```

This operation does not accept a request body.

### 5.4.5.2. Response

#### Example 5.55. Add host: JSON response

```
{
    "aggregate": {
        "availability_zone": "nova",
        "created_at": "2013-08-18T12:17:56.297823",
        "deleted": 0,
        "deleted_at": null,
        "hosts": [
            "21549b2f665945baaa7101926a00143c"
        ],
        "id": 1,
        "metadata": {
            "availability_zone": "nova"
        },
        "name": "name",
        "updated_at": null
    }
}
```

#### Example 5.56. Add host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
<name>name</name>
<availability_zone>nova</availability_zone>
<deleted>False</deleted>
<created_at>2013-08-18 12:17:57.255783</created_at>
```

```
<updated_at>None</updated_at>
<hosts>
  <host>bdb794e5cdd3447c9132ddd8e5cc3d0b</host>
</hosts>
<deleted_at>None</deleted_at>
<id>1</id>
<metadata>
  <availability_zone>nova</availability_zone>
</metadata>
</aggregate>
```

This operation does not return a response body.

## 5.4.6. Create or update aggregate metadata

Method	URI	Description
POST	/v3/os-aggregates/{aggregate_id}/action	Creates or replaces metadata for an aggregate.

Normal response codes: 200

### 5.4.6.1. Request

#### Example 5.57. Create or update aggregate metadata: JSON request

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

#### Example 5.58. Create or update aggregate metadata: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<set_metadata>
    <metadata>
        <key>value</key>
    </metadata>
</set_metadata>
```

This operation does not accept a request body.

### 5.4.6.2. Response

#### Example 5.59. Create or update aggregate metadata: JSON response

```
{
    "aggregate": {
        "availability_zone": "nova",
        "created_at": "2013-08-18T12:17:55.959571",
        "deleted": 0,
        "deleted_at": null,
        "hosts": [],
        "id": 1,
        "metadata": {
            "availability_zone": "nova",
            "key": "value"
        },
        "name": "name",
        "updated_at": null
    }
}
```

#### Example 5.60. Create or update aggregate metadata: XML response

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

```
<aggregate>
  <name>name</name>
  <availability_zone>nova</availability_zone>
  <deleted>False</deleted>
  <created_at>2013-08-18 12:17:56.891817</created_at>
  <updated_at>None</updated_at>
  <hosts/>
  <deleted_at>None</deleted_at>
  <id>1</id>
  <metadata>
    <key>value</key>
    <availability_zone>nova</availability_zone>
  </metadata>
</aggregate>
```

This operation does not return a response body.

## 5.4.7. Remove host

Method	URI	Description
POST	/v3/os-aggregates/{aggregate_id}/action	Removes a host from the specified aggregate.

**Normal response codes:** 202

### 5.4.7.1. Request

#### Example 5.61. Remove host: JSON request

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

#### Example 5.62. Remove host: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<remove_host host="bdb794e5cdd3447c9132ddd8e5cc3d0b" />
```

This operation does not accept a request body.

### 5.4.7.2. Response

#### Example 5.63. Remove host: JSON response

```
{
    "aggregate": {
        "availability_zone": "nova",
        "created_at": "2013-08-18T12:17:56.990581",
        "deleted": 0,
        "deleted_at": null,
        "hosts": [],
        "id": 1,
        "metadata": {
            "availability_zone": "nova"
        },
        "name": "name",
        "updated_at": null
    }
}
```

#### Example 5.64. Remove host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
    <name>name</name>
    <availability_zone>nova</availability_zone>
    <deleted>False</deleted>
    <created_at>2013-08-18 12:17:57.255783</created_at>
    <updated_at>None</updated_at>
    <hosts/>
```

```
<deleted_at>None</deleted_at>
<id>1</id>
<metadata>
  <availability_zone>nova</availability_zone>
</metadata>
</aggregate>
```

This operation does not return a response body.

## 5.5. Cells (os-cells)

Enables cells-related functionality such as adding neighbor cells, listing neighbor cells, and getting the capabilities of the local cell.

Method	URI	Description
<b>GET</b>	/v3/os-cells	Lists cells.
<b>GET</b>	/v3/os-cells	Lists cells with details.
<b>GET</b>	/v3/os-cells/cell13	Shows data for a specified cell.
<b>GET</b>	/v3/os-cells/nova/capacities	Shows capacities for a specified cell.

## 5.5.1. List cells

Method	URI	Description
GET	/v3/os-cells	Lists cells.

When cells are not enabled, the cells API calls return a 501 Not Implemented error.

**Normal response codes:** 200

**Error response codes:** notImplemented (501)

### 5.5.1.1. Request

This operation does not accept a request body.

### 5.5.1.2. Response

#### Example 5.65. List cells: JSON response

```
{
    "cells": [
        {
            "name": "cell1",
            "rpc_host": null,
            "rpc_port": null,
            "type": "child",
            "username": "username1"
        },
        {
            "name": "cell3",
            "rpc_host": null,
            "rpc_port": null,
            "type": "child",
            "username": "username3"
        },
        {
            "name": "cell5",
            "rpc_host": null,
            "rpc_port": null,
            "type": "child",
            "username": "username5"
        },
        {
            "name": "cell2",
            "rpc_host": null,
            "rpc_port": null,
            "type": "parent",
            "username": "username2"
        },
        {
            "name": "cell4",
            "rpc_host": null,
            "rpc_port": null,
            "type": "parent",
            "username": "username4"
        }
    ]
}
```

```
    ]  
}
```

### Example 5.66. List cells: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<cells xmlns="http://docs.rackspacecloud.com/servers/api/v1.0">  
  <cell username="username1" rpc_host="None" type="child" name="cell1"  
    rpc_port="None"/>  
  <cell username="username3" rpc_host="None" type="child" name="cell3"  
    rpc_port="None"/>  
  <cell username="username5" rpc_host="None" type="child" name="cell5"  
    rpc_port="None"/>  
  <cell username="username2" rpc_host="None" type="parent" name="cell2"  
    rpc_port="None"/>  
  <cell username="username4" rpc_host="None" type="parent" name="cell4"  
    rpc_port="None"/>  
</cells>
```

This operation does not return a response body.

## 5.5.2. Lists cells with details

Method	URI	Description
GET	/v3/os-cells	Lists cells with details.

When cells are not enabled, the cells API calls return a 501 Not Implemented error.

**Normal response codes:** 200

**Error response codes:** notImplemented (501)

### 5.5.2.1. Request

This operation does not accept a request body.

### 5.5.2.2. Response

#### Example 5.67. Lists cells with details: JSON response

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

#### Example 5.68. Lists cells with details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<cells xmlns="http://docs.rackspacecloud.com/servers/api/v1.0"/>
```

This operation does not return a response body.

## 5.5.3. Shows cell data

Method	URI	Description
GET	/v3/os-cells/cell3	Shows data for a specified cell.

When cells are not enabled, the cells API calls return a 501 Not Implemented error.

**Normal response codes:** 200

**Error response codes:** notImplemented (501)

### 5.5.3.1. Request

This operation does not accept a request body.

### 5.5.3.2. Response

#### Example 5.69. Shows cell data: JSON response

```
{
  "cell": {
    "name": "cell3",
    "rpc_host": null,
    "rpc_port": null,
    "type": "child",
    "username": "username3"
  }
}
```

#### Example 5.70. Shows cell data: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<cell xmlns="http://docs.rackspacecloud.com/servers/api/v1.0" username=
"username3" rpc_host="None" type="child" name="cell3" rpc_port="None"/>
```

This operation does not return a response body.

## 5.5.4. Shows cell capacities

Method	URI	Description
GET	/v3/os-cells/nova/capacities	Shows capacities for a specified cell.

When cells are not enabled, the cells API calls return a 501 Not Implemented error.

**Normal response codes:** 200

**Error response codes:** notImplemented (501)

### 5.5.4.1. Request

This operation does not accept a request body.

### 5.5.4.2. Response

#### Example 5.71. Shows 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
        }
      }
    }
  }
}
```

#### Example 5.72. Shows cell capacities: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<cell xmlns="http://docs.rackspacecloud.com/servers/api/v1.0">
  <capacities>
    <ram_free total_mb="7680">
      <unit_by_mb unit="0" mb="8192"/>
      <unit_by_mb unit="13" mb="512"/>
      <unit_by_mb unit="1" mb="4096"/>
    
```

```
<unit_by_mb unit="3" mb="2048" />
<unit_by_mb unit="0" mb="16384" />
</ram_free>
<disk_free total_mb="1052672">
    <unit_by_mb unit="11" mb="81920" />
    <unit_by_mb unit="46" mb="20480" />
    <unit_by_mb unit="5" mb="163840" />
    <unit_by_mb unit="23" mb="40960" />
    <unit_by_mb unit="0" mb="0" />
</disk_free>
</capacities>
</cell>
```

This operation does not return a response body.

## 5.6. Root certificates (os-certificates)

Creates and shows details for a root certificate.

Method	URI	Description
<b>POST</b>	/v3/os-certificates	Creates a certificate.
<b>GET</b>	/v3/os-certificates/root	Shows details for a specified certificate.

## 5.6.1. Create certificate

Method	URI	Description
POST	/v3/os-certificates	Creates a certificate.

**Normal response codes:** 201

### 5.6.1.1. Request

This operation does not accept a request body.

### 5.6.1.2. Response

#### Example 5.73. Create certificate: JSON response

```
{
    "certificate": {
        "data": "Certificate:\n      Data:\n          Version: 1 (0x0)\nSerial Number: 1018 (0x3fa)\n      Signature Algorithm: md5WithRSAEncryption\n\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\n      Subject Public Key Info:\n          Public Key Algorithm:\n              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\n          Exponent: 65537 (0x10001)\n          Signature Algorithm: md5WithRSAEncryption\n\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\nnVDEWMBQGA1UEBxMNTW91bnRhaW4gVm1ldzETMBEGA1UECBMKQ2FsaWZvcn5pYTEL\nnMAkGA1UEBhMCVVMwHhcNMTMwODEyMDcyMDMwWhcNMTQwODEyMDcyMDMwWjB2MQsw\nnCQYDVQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcn5pYTEsMBAGA1UECgwJT3B1b1N0\nnYWNrMRAwDgYDVQQLDA0b3ZhRGV2MSwwKgYDVQQDCNvcGVuc3RhY2stZmFrZS0y\nnMDEzLTA4LTEyVDA30jIwOjMwWjCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYE\nnrP+x0e1UTjVsNLSPCwRQJaPiTwJMTyZZvfP969oYwjajqY0JyH4hPOuznn45EKtO4\nn1HsgQfhIAleRTBZi8SHU8kC1h1DZYfC+/9iNn0uqagc4on+HIfzmbh0K1RqQDmDC\nnJOmO6Ggb6fPGsHzaxSBmm4Xq9cmn3u4WsVGgTeOVmN8CAwEAATANBgkqhkiG9w0B\nnAQFQAAOBgQAVQspxzDKv3M9Fkd+KuDDef3iApyXC2YE+s90izDv4lOePBPaTBJ6F\nn1BBA/1oHRyS1rpOtjeHmVEqNs1TxI0Eawv2rzh4AsUFGYmCLbr9ETweGMkMPQOT\nnbrxmcDTuA3iKHT1k6CAvkIGOSR0HNxVmQstYOa1WzulHxngLDnUpyg==\n-----END CERTIFICATE-----\n",
```

```

    "private_key": "-----BEGIN RSA PRIVATE KEY-----\\nMIICXgIBAAKBgQC.../7HR7VRONWw0tI8LBFAlo+JPAkxPJ1m98/3r2hjCNqpjQnIf\\niE867OefjkQq07iUeyBB+EgCV5FMFmLxIdTyQLWGUNlh8L7/2I2fS6pqBziif4ch\\n/OZuHQqVGpAOYMIk6Y7oaBvp88awfNrFIGabher1yafe7haxUaBN45WY3wIDAQAB\\nAoGBAIrcr2I/KyWf0hw4Nn10V9TuyE/9Gz2JHg3QFKjFJox2DqygADT5WAeHc6Bq\\nNKnf0NA2SL1LSpm+ql01tv0w4VjE5TF6OHiIzHuTTnXggG6vuA8rxp6L24HtkAcc\\n0CBno9ggSX6jVornJPBfxpkwITYsvH57BUFVD7ovbPyWGzS5AkEA1JeUtL6zxwps\\nWRrlaJ8I112uQk/RUIvSZOU61s+B190zvHikFy8LD8CI6vvBmjC/IzuZVedufjq...\\n4vX82uDO3QJBANBSH2b2dyB4AGVFY9vXMRtALAspJHbLHy+zTKx1GPFIuz7Se3ps\\n8Kehz4C/CBXgQkk194dwFSGE19/PQfyJROSCQQCCFDJZhrtBUMwMZ2zSRiN5BUGt\\nbwuncS+OS1Su3Yz5VRYq2BZYEPHKtYrAfklWQ8eRwTaWaN5pFE/fb380gQXdAkA4\\nDm0W/K0z1HbuyUxEpNQ28/6mBi0ktiWvLT0tioq6sYmXLwZA/D2JrhXrG/xt/o13\\nr8jqrfrNsLByLhAgh0N/AkEA12eR0O971TEgFNqzIQwVmIAm9mBO3cnf3tycv1DU\\nm6eb2CS242y4QalfCCAEjxoJURdfsm3/D1iFo00X+IWF+A==\\n-----END RSA PRIVATE\\nKEY-----\\n"
    }
}

```

### Example 5.74. Create certificate: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<certificate private_key="-----BEGIN RSA PRIVATE KEY-----&#10;MIICXgIBAAKBgQC4UMjbmUFkFPFSM0HRz6ULcpwmg2eaJ+EaIwhLrSyghPy1NxL&#10;#RbS5gurxDZahspmtULNOUA6YqCAMn0t1EOZe3vWAnckAvnKMZu146gtBfkxTFig4&#10;#P9P1LWgyo00cSaV2T8Itrb2ZV90QAlAlrF51zhXUvW63jjyv51PsM2PhWQIDAQAB&#10;#AoGBAKE7Qcke02/9/9m3SmsQf0r+3fTMPheRK+Fw1yX7xmPU+ABXBfHuox4Ykm6W&#10;#9WqzYk1DJ+bCjzgo8gg4nHhmdS2RIqNeWczOaOHprWzPVEbwaR10sSuLUYOBfuko&#10;#brvPwgw856UcsByGdM5Dif5DhtWtR3znY4UE0I3tzKelyMrRAkEA6wdpqdFoHjhD&#10;#80GRic2ZHmvxwi2wKZ0izf29vSvRunrYxH79zADZqiIvs90cCuYjdJT4JJYfzU8&#10;#cm316QJBbQJBAMjC9ur60uoxzM7cUsOfiHqrQvhuy3WYkkMcqBNcms0Z2G7cRqxF&#10;#ZEroqNyXHUiN3WXbi/pRsdnMX8sta7QZWB0CQQDQiaVBnqLILHqsMx1UQVFJYvIC&#10;#H1PduKV3LRu7zDNRAdtwj1VaDvoVjpi1dHPWS3hy7hqgem+3f18BEW4cT3oJAkBp&#10;#+wQdtTTykIK21GJp7vaivF8hp892Suj2v3GwOTXAw3r1Iv/Q6HiCfCD+RvsdMAY&#10;#PS/16H6Qa1BafFzdIh+tAkEAsAb0Y4zN5TACTOPBj2Payhgo7rkqdM0tsR5pTE4f&#10;#Wx7gqKn7vvq9CJ5Lp80RzhqIbf4hC9wEhjHHomQ43AjK3Q==#10;-----END RSA PRIVATE\\nKEY-----#10;" data="Certificate:#10; Data:#10; Version: 1\\n(0x0)#10; Serial Number: 1019 (0x3fb)#10; Signature Algorithm:\\nmd5WithRSAEncryption#10; Issuer: O=NOVA ROOT, L=Mountain View,\\nST=California, C=US#10; Validity#10; Not Before: Aug\\n12 07:20:30 2013 GMT#10; Not After : Aug 12 07:20:30 2014\\nGMT#10; Subject: C=US, ST=California, O=OpenStack, OU=NovaDev,\\nCN=openstack-fake-2013-08-12T07:20:30Z#10; Subject Public Key\\nInfo:#10; Public Key Algorithm: rsaEncryption#10;\\nPublic-Key: (1024 bit)#10; Modulus:#10;\\n00:b8:50:c8:db:98:55:1f:90:53:c5:48:cd:07:47:#10;\\n3e:94:2d:ca:70:9a:0d:9e:68:9f:84:68:8c:21:2e:#10;\\nb4:b2:82:13:f2:d4:dc:4b:45:b4:b9:82:ea:f1:0d:#10;\\n96:a1:b2:99:ad:50:b3:4e:50:0e:98:a8:20:0c:9f:#10;\\n4b:75:10:e6:5e:de:f5:80:9d:c9:00:be:72:8c:66:#10;\\ne9:78:ea:0b:41:7e:4c:53:14:88:38:3f:d3:f5:2d:#10;\\n68:32:a0:ed:1c:49:a5:76:4f:c2:2d:ad:bd:99:57:#10;\\nd3:90:02:50:25:ac:5e:75:ce:15:d4:bd:6e:b7:8e:#10;\\n3c:af:e7:53:ec:33:63:e1:59:#10; Exponent:\\n65537 (0x10001)#10; Signature Algorithm: md5WithRSAEncryption&#10;\\n5e:d0:d5:7a:4c:85:02:7d:a4:11:98:97:ee:9a:57:91:14:de:&#10;\\nf7:16:6b:1b:2d:93:57:13:4c:50:5a:27:13:1c:3a:f0:72:12:&#10;\\n1c:97:0a:45:be:30:25:ee:25:30:ac:3d:b3:81:b2:ca:9c:a5:&#10;\\ncd:5a:2a:35:21:c6:98:3a:2b:eb:27:bf:88:4a:aa:69:f5:5b:&#10;\\nd4:06:00:6d:ce:d6:69:2e:75:fe:6e:f2:36:c7:52:59:9a:0b:&#10;\\ne5:0a:63:14:a3:f1:06:36:95:04:76:94:ba:1a:44:9f:84:1d:&

```

```
#10; f2:f9:18:c6:e6:35:ff:75:35:5c:18:ef:35:fb:d7:9c:87:5a:&
#10; 73:0c"#10;-----BEGIN CERTIFICATE-----&
#10;MIICNDCCAZ0CAgP7MA0GCSqGSIb3DQEBAUAME4xEjAQBgNVBAoTCU5PVkEgUk9P&
#10;VDEWMBQGA1UEBxMNTW91bnRhaW4gVmlldzETMBEGA1UECBMKQ2FsaWZvcm5pYTEL&
#10;MAkGA1UEBhMCVVMwHhcNMTMwODEyMDcyMDMwWhcNMTQwODEyMDcyMDMwWjb2MQsw&
#10;CQYDVQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcm5pYTESMBAGA1UECgwJT3B1b1N0&
#10;YWNrMRAwDgYDVQQLDA0b3ZhRGV2MSwwKgYDVQQDCNvcGVuc3RhY2stZmFrZS0y&
#10;MDEzLTA4LTEyVDA3OjIwOjMwWjCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAA&
#10;uFDI25hVH5BTxUjNB0c+1C3KcJoNmifhGiMIS6soIT8tTcS0W0uYLq8Q2WobKZ&
#10;rVCzT1AOmKggDJ9LdRDmXt71gJ3JAL5yjGbpeOoLQX5MUxSIOD/T9S1oMqDtHEml&
#10;dk/CLa29mVfTkAJQJaxedc4V1L1ut448r+dT7DNj4VkcAwEAATANBgkqhkiG9w0B&
#10;AQQFAOBgQBe0NV6TIUCfaQRmJfumleRFN73FmsbLZNXE0xQWicthDrwchIclwpF&
#10;vjAl7iUwrD2zgbLKnKXNWio1IcaYOivrJ7+ISqpp9VvUBgBtzlZpLnX+bvI2x1JZ&
#10;mgvlCmMUo/EGnpUEDpS6GkSfhB3y+RjG5jX/dTvcGO81+9ech1pzDA=="#10;-----END
CERTIFICATE-----&#10;"/>
```

This operation does not return a response body.

## 5.6.2. Show certificate details

Method	URI	Description
GET	/v3/os-certificates/root	Shows details for a specified certificate.

**Normal response codes:** 200

### 5.6.2.1. Request

This operation does not accept a request body.

### 5.6.2.2. Response

#### Example 5.75. Show certificate details: JSON response

```
{
  "certificate": {
    "data": "-----BEGIN CERTIFICATE-----\\nMIICyZCCAjSgAwIBAgIJAJ8zS1xUp/m4MA0GCSqGSIb3DQEBAUAME4xEjAQBgNV\\nBAoTCU5PVkEgUk9PVDEWMBQGA1UEBxMNTW91bnRhaW4gVm1ldzETMBEGA1UECBMK\\nQ2FsaWZvcm5pYTELMAKGA1UEBhMCVVMwHhcNMTIxMDE3MDEzMzM5WhcNMTMxMDE3\\nMDEzMzM5WjBOMRIwEAYDVQQKEw1OT1ZBIFJPT1QxFjaUBgNVBAcTDU1vdW50YWlu\\nIFZpZXcxEzARBgNVBAgTCkNhbGlmb3JuaWEExCzAJBgNVBAYTA1VTMIGfMA0GCSqG\\nSib3DQEBAQUAA4GNADCBiQKBgQDXW4QfQQxJG4MqrqK8nU/Lge0mfNKxXj/Gwvg\\n2sQVwxzmKfoxih8Nn6yt0yHNMjhoji1UoWI03TXUnPZRAZmsypGKZeBd7Y1ZOCPB\\nXGZVGrQm+PB2kZU+3cD8fVKcueMLLeZ+Lrt5d0njqM1fFPimHMba4OL6\\nTnPQIDAQABo4GwMIGtMAwGA1UdEwQFMAMBAf8wHQYDVR0OBBYEFKyoKu4SMOFM\\ngx5Ec7p0nrCkabvxMH4GA1UdIwR3MHWAFKy0Ku4SMOFMgx5Ec7p0nrCkabvxoVKk\\nUDBOMRIwEAYDVQQKEw1OT1ZBIFJPT1QxFjaUBgNVBAcTDU1vdW50YWluIFZpZXcx\\neZARBgNVBAgTCkNhbGlmb3JuaWEExCzAJBgNVBAYTA1VTggkAnzNIjFSn+bgwDQYJ\\nKoZIhvcNAQEEBQADgYEAXuvXlu1o/SVvykSLhHW8QiAY00yzN/eDzYmZGomgiuoO/n\\x+ayVzbrz1UWZnBD+1C4h112iELSmf22LjLoF+s/9NyPqHxGL3FrFatBkndaiF8\\nAx/TMEyCP17IQWi+3zzatqOKHSHiG7a9SGn/7o2aNtIKVulfy5GvmlbBjBM/0UE=\n-----END CERTIFICATE-----\\n",
    "private_key": null
  }
}
```

#### Example 5.76. Show certificate details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<certificate private_key="None" data="-----BEGIN CERTIFICATE-----&#10;MIICyZCCAjSgAwIBAgIJAJ8zS1xUp/m4MA0GCSqGSIb3DQEBAUAME4xEjAQBgNV&#10;#BAoTCU5PVkEgUk9PVDEWMBQGA1UEBxMNTW91bnRhaW4gVm1ldzETMBEGA1UECBMK&#10;#Q2FsaWZvcm5pYTELMAKGA1UEBhMCVVMwHhcNMTIxMDE3MDEzMzM5WhcNMTMxMDE3&#10;#MDEzMzM5WjBOMRIwEAYDVQQKEw1OT1ZBIFJPT1QxFjaUBgNVBAcTDU1vdW50YWlu&#10;#IFZpZXcxEzARBgNVBAgTCkNhbGlmb3JuaWEExCzAJBgNVBAYTA1VTMIGfMA0GCSqG&#10;#Sib3DQEBAQUAA4GNADCBiQKBgQDXW4QfQQxJG4MqrqK8nU/Lge0mfNKxXj/Gwvg&#10;#2sQVwxzmKfoxih8Nn6yt0yHNMjhoji1UoWI03TXUnPZRAZmsypGKZeBd7Y1ZOCPB&#10;#XGZVGrQm+PB2kZU+3cD8fVKcueMLLeZ+Lrt5d0njqM1fFPimHMba4OL6&#10;#TnPQIDAQABo4GwMIGtMAwGA1UdEwQFMAMBAf8wHQYDVR0OBBYEFKyoKu4SMOFM&#10;#gx5Ec7p0nrCkabvxMH4GA1UdIwR3MHWAFKy0Ku4SMOFMgx5Ec7p0nrCkabvxoVKk&#10;#UDBOMRIwEAYDVQQKEw1OT1ZBIFJPT1QxFjaUBgNVBAcTDU1vdW50YWluIFZpZXcx&#10;#EzARBgNVBAgTCkNhbGlmb3JuaWEExCzAJBgNVBAYTA1VTggkAnzNIjFSn+bgwDQYJ&#10;#KoZIhvcNAQEEBQADgYEAXuvXlu1o/SVvykSLhHW8QiAY00yzN/eDzYmZGomgiuoO&#10;#x+ayVzbrz1UWZnBD+1C4h112iELSmf22LjLoF+s/9NyPqHxGL3FrFatBkndaiF8&#10;#Ax/
```

```
TMEyCP17IQWi+3zzatqOKHSHiG7a9SGn/7o2aNTIWKVulfy5GvmbBjBM/0UE=&#10;-----END  
CERTIFICATE-----&#10; "/>
```

This operation does not return a response body.

## 5.7. Configuration drive (os-config-drive)

Returns server details for a specific service ID or user.

Method	URI	Description
<b>GET</b>	/v3/servers/{server_id}	Shows details for a specified server.
<b>GET</b>	/v3/servers/detail	Lists server details for a specified user.

## 5.7.1. Show server details

Method	URI	Description
GET	/v3/servers/{server_id}	Shows details for a specified server.

**Normal response codes:** 200

### 5.7.1.1. Response

#### Example 5.77. Show server details: JSON response

```
{
  "server": {
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "mac_addr": "aa:bb:cc:dd:ee:ff",
          "type": "fixed",
          "version": 4
        }
      ]
    },
    "created": "2013-09-22T02:33:23Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "host_id": "1642bbdbd61a0f1c513b4bb6e418326103172698104bfa278eca106b",
    "id": "7838ff1b-b71f-48b9-91e9-7c08de20b249",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    },
    "key_name": null,
    "links": [
      {
        "href": "http://openstack.example.com/v3/servers/7838ff1b-
b71f-48b9-91e9-7c08de20b249",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/servers/7838ff1b-
b71f-48b9-91e9-7c08de20b249",
        "rel": "bookmark"
      }
    ]
  }
}
```

```
        ],
        "metadata": {
            "My Server Name": "Apache1"
        },
        "name": "new-server-test",
        "os-config-drive:config_drive": "",
        "progress": 0,
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2013-09-22T02:33:25Z",
        "user_id": "fake"
    }
}
```

### Example 5.78. Show server details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:os-config-drive="http://docs.openstack.org/compute/
ext/config_drive/api/v3" xmlns:atom="http://www.w3.org/2005/Atom"
         xmlns="http://docs.openstack.org/compute/api/v1.1" status="ACTIVE"
         updated="2013-09-22T02:33:23Z" user_id="fake" name="new-server-test"
         created="2013-09-22T02:33:21Z" tenant_id="openstack" progress="0"
         host_id="c45cc50e4626bbe902a4356530b22f309552027c6ccf7d20bd1f7c43" id=
"e516a1b5-0081-4d8c-86d5-c05b93501076" os-config-drive:config_drive=""
         key_name="None">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" type="fixed" addr="192.168.0.3" mac_addr=
"aa:bb:cc:dd:ee:ff"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v3/servers/
e516a1b5-0081-4d8c-86d5-c05b93501076" rel="self"/>
    <atom:link href="http://openstack.example.com/servers/
e516a1b5-0081-4d8c-86d5-c05b93501076" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 5.7.2. List server details for a user

Method	URI	Description
GET	/v3/servers/detail	Lists server details for a specified user.

**Normal response codes:** 200

### 5.7.2.1. Response

#### Example 5.79. List server details for a user: JSON response

```
{
    "servers": [
        {
            "addresses": {
                "private": [
                    {
                        "addr": "192.168.0.3",
                        "mac_addr": "aa:bb:cc:dd:ee:ff",
                        "type": "fixed",
                        "version": 4
                    }
                ]
            },
            "created": "2013-09-22T02:33:17Z",
            "flavor": {
                "id": "1",
                "links": [
                    {
                        "href": "http://openstack.example.com/flavors/1",
                        "rel": "bookmark"
                    }
                ]
            },
            "host_id": "1ed067c90341cd9d94bbe5da960922b56f107262cdc75719a0d97b78",
            "id": "f0318e69-11eb-4aed-9840-59b6c72beee8",
            "image": {
                "id": "70a599e0-31e7-49b7-b260-868f441e862b",
                "links": [
                    {
                        "href": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                        "rel": "bookmark"
                    }
                ]
            },
            "key_name": null,
            "links": [
                {
                    "href": "http://openstack.example.com/v3/servers/
f0318e69-11eb-4aed-9840-59b6c72beee8",
                    "rel": "self"
                },
                {

```

```

        "href": "http://openstack.example.com/servers/
f0318e69-11eb-4aed-9840-59b6c72beee8",
        "rel": "bookmark"
    }
],
"metadata": {
    "My Server Name": "Apache1"
},
"name": "new-server-test",
"os-config-drive:config_drive": "",
"progress": 0,
"status": "ACTIVE",
"tenant_id": "openstack",
"updated": "2013-09-22T02:33:19Z",
"user_id": "fake"
}
]
}

```

### Example 5.80. List server details for a user: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:os-config-drive="http://docs.openstack.org/compute/ext/
config_drive/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2013-09-22T02:33:20Z"
    user_id="fake" name="new-server-test" created=
    "2013-09-22T02:33:18Z" tenant_id="openstack" progress="0" host_id=
    "bdabf9c023874e6a5a3f6dc032222e7967707e577765f16e2836fb0b" id="lfa6f088-
dece-466e-9fe4-7e01645852ba" os-config-drive:config_drive="" key_name="None">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/
>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" type="fixed" addr="192.168.0.3" mac_addr=
                "aa:bb:cc:dd:ee:ff"/>
            </network>
        </addresses>
        <atom:link href="http://openstack.example.com/v3/servers/lfa6f088-
dece-466e-9fe4-7e01645852ba" rel="self"/>
        <atom:link href="http://openstack.example.com/servers/lfa6f088-
dece-466e-9fe4-7e01645852ba" rel="bookmark"/>
    </server>
</servers>

```

This operation does not return a response body.

## 5.8. Server deferred delete (os-deferred-delete)

Force-deletes a server or restores a deleted server.

Method	URI	Description
<b>POST</b>	/v3/servers/{server_id}/action	Force deletes an instance before deferred cleanup.
<b>POST</b>	/v3/servers/{server_id}/action	Restores a previously deleted instance.

## 5.8.1. Force delete instance

Method	URI	Description
POST	/v3/servers/{server_id}/action	Force deletes an instance before deferred cleanup.

**Normal response codes:** 202

### 5.8.1.1. Request

#### Example 5.81. Force delete instance: JSON request

```
{  
    "force_delete": null  
}
```

#### Example 5.82. Force delete instance: XML request

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

This operation does not accept a request body.

## 5.8.2. Restore deleted instance

Method	URI	Description
POST	/v3/servers/{server_id}/action	Restores a previously deleted instance.

**Normal response codes:** 202

### 5.8.2.1. Request

#### Example 5.83. Restore deleted instance: JSON request

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

#### Example 5.84. Restore deleted instance: XML request

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

This operation does not accept a request body.

## 5.9. Evacuate (os-evacuate)

Enables server evacuation.

Method	URI	Description
POST	/v3/servers/{server_id}/action	Evacuates a server from a failed host to a new one.

## 5.9.1. Evacuate server

Method	URI	Description
POST	/v3/servers/{server_id}/action	Evacuates a server from a failed host to a new one.

Normal response codes: 200

### 5.9.1.1. Request

#### Example 5.85. Evacuate server: JSON request

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

#### Example 5.86. Evacuate server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<evacuate xmlns="http://docs.openstack.org/compute/api/v2"
  host="c8b85acac4394623a8bd99f66e6a623d"
  admin_password="MySecretPass"
  on_shared_storage="False"/>
```

This operation does not accept a request body.

### 5.9.1.2. Response

#### Example 5.87. Evacuate server: JSON response

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

#### Example 5.88. Evacuate server: XML response

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

This operation does not return a response body.

## 5.10. Servers with extended availability zones (os-extended-availability-zone)

Shows the instance availability zone for compute nodes (nova-compute). Internal services appear in their own internal availability zone.

Method	URI	Description
GET	/v3/servers/{server_id}	Shows details for a specified server.

Method	URI	Description
<b>GET</b>	/v3/servers/detail	Lists server details for a specified user.

## 5.10.1. Show server details

Method	URI	Description
GET	/v3/servers/{server_id}	Shows details for a specified server.

**Normal response codes:** 200

### 5.10.1.1. Response

#### Example 5.89. Show server details: JSON response

```
{
  "server": {
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "mac_addr": "aa:bb:cc:dd:ee:ff",
          "type": "fixed",
          "version": 4
        }
      ]
    },
    "created": "2013-09-16T02:54:56Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "host_id": "b75d6736650f9b272223ceb48f4cde001de1856e381613a922117ab7",
    "id": "f22e4521-d03a-4e9f-9fd3-016b9e227219",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    },
    "key_name": null,
    "links": [
      {
        "href": "http://openstack.example.com/v3/servers/f22e4521-
d03a-4e9f-9fd3-016b9e227219",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/servers/f22e4521-
d03a-4e9f-9fd3-016b9e227219",
        "rel": "bookmark"
      }
    ]
  }
}
```

```
        ],
        "metadata": {
            "My Server Name": "Apache1"
        },
        "name": "new-server-test",
        "os-extended-availability-zone:availability_zone": "nova",
        "progress": 0,
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2013-09-16T02:54:57Z",
        "user_id": "fake"
    }
}
```

### Example 5.90. Show server details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:os-extended-availability-zone="http://docs.openstack.org/
compute/ext/extended_availability_zone/api/v3" xmlns:atom="http://www.w3.
org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" status=
"ACTIVE" updated="2013-09-16T02:55:02Z" user_id="fake" name="new-server-test"
created="2013-09-16T02:55:01Z" tenant_id="openstack" progress="0" host_id=
"146f5f54649a2655d2daed87b2da7eec04777520ce4aaa0642845173" id="d693cfcb-
b6b8-4d9c-acb9-4db3bd16593f" os-extended-availability-zone:availability_zone=
"nova" key_name="None">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" type="fixed" addr="192.168.0.3" mac_addr=
"aa:bb:cc:dd:ee:ff"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v3/servers/d693cfcb-b6b8-4d9c-
acb9-4db3bd16593f" rel="self"/>
    <atom:link href="http://openstack.example.com/servers/d693cfcb-b6b8-4d9c-
acb9-4db3bd16593f" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 5.10.2. List server details for user

Method	URI	Description
GET	/v3/servers/detail	Lists server details for a specified user.

**Normal response codes:** 200

### 5.10.2.1. Response

#### Example 5.91. List server details for user: JSON response

```
{
  "servers": [
    {
      "addresses": {
        "private": [
          {
            "addr": "192.168.0.3",
            "mac_addr": "aa:bb:cc:dd:ee:ff",
            "type": "fixed",
            "version": 4
          }
        ]
      },
      "created": "2013-09-16T02:54:56Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "host_id": "cf5540800371e53064a60b36ff9d6d1d6a8719ffc870c63a270c6bee",
      "id": "55f43fa2-dc7c-4c0b-b21a-76f9abe516f9",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
          {
            "href": "http://glance.openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
            "rel": "bookmark"
          }
        ]
      },
      "key_name": null,
      "links": [
        {
          "href": "http://openstack.example.com/v3/servers/55f43fa2-dc7c-4c0b-b21a-76f9abe516f9",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/servers/55f43fa2-dc7c-4c0b-b21a-76f9abe516f9",
          "rel": "bookmark"
        }
      ]
    }
  ]
}
```

```

        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "os-extended-availability-zone:availability_zone": "nova",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2013-09-16T02:54:58Z",
    "user_id": "fake"
}
]
}

```

### Example 5.92. List server details for user: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:os-extended-availability-zone="http://docs.openstack.org/
compute/ext/extended_availability_zone/api/v3" xmlns:atom="http://www.w3.org/
2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2013-09-16T02:55:00Z"
    user_id="fake" name="new-server-test" created=
    "2013-09-16T02:54:59Z" tenant_id="openstack" progress="0" host_id=
    "1c938f2d8fe831c1a6f1e1c013ae40400218c92013d1452fe4a8fffc3" id="81a6e93a-
    eeeb-4169-b562-bbe42bffa27" os-extended-availability-zone:availability_zone=
    "nova" key_name="None">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/
>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" type="fixed" addr="192.168.0.3" mac_addr=
                "aa:bb:cc:dd:ee:ff"/>
            </network>
        </addresses>
        <atom:link href="http://openstack.example.com/v3/servers/81a6e93a-
        eeeb-4169-b562-bbe42bffa27" rel="self"/>
        <atom:link href="http://openstack.example.com/servers/81a6e93a-eee-
        b-4169-b562-bbe42bffa27" rel="bookmark"/>
    </server>
</servers>

```

This operation does not return a response body.

## 5.11. Server extended attributes (os-extended-server-attributes)

Shows metadata for servers.

Method	URI	Description
<b>GET</b>	/v3/servers	Shows details for a specified server.
<b>GET</b>	/v3/servers/details	Lists server details for a specified user.

## 5.11.1. Shows server details

Method	URI	Description
GET	/v3/servers	Shows details for a specified server.

**Normal response codes:** 200

### 5.11.1.1. Response

#### Example 5.93. Shows server details: JSON response

```
{
  "server": {
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "mac_addr": "aa:bb:cc:dd:ee:ff",
          "type": "fixed",
          "version": 4
        }
      ]
    },
    "created": "2013-09-16T02:55:07Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "host_id": "3bf189131c61d0e71b0a8686a897a0f50d1693b48c47b721fe77155b",
    "id": "c278163e-36f9-4cf2-b1ac-80db4c63f7a8",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    },
    "key_name": null,
    "links": [
      {
        "href": "http://openstack.example.com/v3/servers/
c278163e-36f9-4cf2-b1ac-80db4c63f7a8",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/servers/
c278163e-36f9-4cf2-b1ac-80db4c63f7a8",
        "rel": "bookmark"
      }
    ]
  }
}
```

```
        ],
        "metadata": {
            "My Server Name": "Apache1"
        },
        "name": "new-server-test",
        "os-extended-server-attributes:host":
"c5f474bf81474f9dbbc404d5b2e4e9b3",
        "os-extended-server-attributes:hypervisor_hostname": "fake-mini",
        "os-extended-server-attributes:instance_name": "instance-00000001",
        "progress": 0,
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2013-09-16T02:55:08Z",
        "user_id": "fake"
    }
}
```

### Example 5.94. Shows server details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:os-extended-server-attributes="http://docs.openstack.org/
compute/ext/extended_server_attributes/api/v3" xmlns:atom="http://www.w3.
org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" status=
"ACTIVE" updated="2013-09-16T02:54:58Z" user_id="fake" name="new-server-
test" created="2013-09-16T02:54:56Z" tenant_id="openstack" progress=
"0" host_id="6a5d92e002c044ec7a57b04d7b564d6536d172601518d66559bfd736"
id="17f2690d-2369-4b8a-9b28-66e503c079e9" os-extended-server-
attributes:hypervisor_hostname="fake-mini" os-extended-server-
attributes:instance_name="instance-00000001" os-extended-server-
attributes:host="45dc15db04d54001801b10996b49db9e" key_name="None">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" type="fixed" addr="192.168.0.3" mac_addr=
"aa:bb:cc:dd:ee:ff"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v3/servers/
17f2690d-2369-4b8a-9b28-66e503c079e9" rel="self"/>
    <atom:link href="http://openstack.example.com/servers/
17f2690d-2369-4b8a-9b28-66e503c079e9" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 5.11.2. Lists server details for user

Method	URI	Description
GET	/v3/servers/details	Lists server details for a specified user.

**Normal response codes:** 200

### 5.11.2.1. Response

#### Example 5.95. Lists server details for user: JSON response

```
{
  "servers": [
    {
      "addresses": {
        "private": [
          {
            "addr": "192.168.0.3",
            "mac_addr": "aa:bb:cc:dd:ee:ff",
            "type": "fixed",
            "version": 4
          }
        ]
      },
      "created": "2013-09-16T02:55:03Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "host_id": "63cf07a9fd82e1d2294926ec5c0d2e1e0ca449224246df75e16f23dc",
      "id": "a8c1c13d-ec7e-47c7-b4ff-077f72c1ca46",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
          {
            "href": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
            "rel": "bookmark"
          }
        ]
      },
      "key_name": null,
      "links": [
        {
          "href": "http://openstack.example.com/v3/servers/a8c1c13d-
ec7e-47c7-b4ff-077f72c1ca46",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/servers/a8c1c13d-
ec7e-47c7-b4ff-077f72c1ca46",
        }
      ]
    }
  ]
}
```

```

        "rel": "bookmark"
    }
],
"metadata": {
    "My Server Name": "Apache1"
},
"name": "new-server-test",
"os-extended-server-attributes:host":
"bc8efef4fdb7148a4bb921a2b03d17de6",
"os-extended-server-attributes:hypervisor_hostname": "fake-mini",
"os-extended-server-attributes:instance_name":
"instance-00000001",
"progress": 0,
"status": "ACTIVE",
"tenant_id": "openstack",
"updated": "2013-09-16T02:55:05Z",
"user_id": "fake"
}
]
}
}

```

### Example 5.96. Lists server details for user: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:os-extended-server-attributes="http://docs.openstack.org/
compute/ext/extended_server_attributes/api/v3" xmlns:atom="http://www.w3.org/
2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2013-09-16T02:55:11Z"
    user_id="fake" name="new-server-test" created=
"2013-09-16T02:55:10Z" tenant_id="openstack" progress="0" host_id=
"b9389d025797d258fd68840325fd2905a5f004d01d0f3dea37691c27"
    id="cf2daa06-ae23-430c-9c6e-6ea303339bd8" os-extended-server-
    attributes:hypervisor_hostname="fake-mini" os-extended-server-
    attributes:instance_name="instance-00000001" os-extended-server-
    attributes:host="dlba3610704b49fcfb70477d4d5840cd" key_name="None">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/
>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" type="fixed" addr="192.168.0.3" mac_addr=
"aa:bb:cc:dd:ee:ff"/>
            </network>
        </addresses>
        <atom:link href="http://openstack.example.com/v3/servers/cf2daa06-
ae23-430c-9c6e-6ea303339bd8" rel="self"/>
        <atom:link href="http://openstack.example.com/servers/cf2daa06-
ae23-430c-9c6e-6ea303339bd8" rel="bookmark"/>
    </server>
</servers>

```

This operation does not return a response body.

## 5.12. Server extended status (`os-extended-status`)

Shows extended status information, `vm_state`, `task_state`, and `power_state`, in detailed server responses.

Method	URI	Description
<b>GET</b>	<code>/v3/servers/{server_id}</code>	Shows details for a specified server.
<b>GET</b>	<code>/v3/servers/detail</code>	Lists servers with details for a specified user.

## 5.12.1. Show server details

Method	URI	Description
GET	/v3/servers/{server_id}	Shows details for a specified server.

**Normal response codes:** 200

### 5.12.1.1. Response

#### Example 5.97. Show server details: JSON response

```
{
  "server": {
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "mac_addr": "aa:bb:cc:dd:ee:ff",
          "type": "fixed",
          "version": 4
        }
      ]
    },
    "created": "2013-09-16T03:07:06Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "host_id": "46d2aa2d637bd55606304b611a1928627ee1278c149aef2206268d6e",
    "id": "a868cb5e-c794-47bf-9cd8-e302b72bb94b",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    },
    "key_name": null,
    "links": [
      {
        "href": "http://openstack.example.com/v3/servers/a868cb5e-
c794-47bf-9cd8-e302b72bb94b",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/servers/a868cb5e-
c794-47bf-9cd8-e302b72bb94b",
        "rel": "bookmark"
      }
    ]
  }
}
```

```
        ],
        "metadata": {
            "My Server Name": "Apache1"
        },
        "name": "new-server-test",
        "os-extended-status:locked_by": null,
        "os-extended-status:power_state": 1,
        "os-extended-status:task_state": null,
        "os-extended-status:vm_state": "active",
        "progress": 0,
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2013-09-16T03:07:07Z",
        "user_id": "fake"
    }
}
```

### Example 5.98. Show server details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:os-extended-status="http://docs.openstack.org/compute/
ext/extended_status/api/v3" xmlns:atom="http://www.w3.org/2005/Atom"
         xmlns="http://docs.openstack.org/compute/api/v1.1" status="ACTIVE"
         updated="2013-09-16T03:07:19Z" user_id="fake" name="new-server-test"
         created="2013-09-16T03:07:17Z" tenant_id="openstack" progress="0"
         host_id="9d03fc2b812a0c9b5b170f9b6eb3eef2f5e0f4829balc1d1be3e3c35" id=
"348a9852-69c2-4dc2-a78d-fe7af4c2bf19" os-extended-status:vm_state="active"
         os-extended-status:locked_by="None" os-extended-status:power_state="1" os-
extended-status:task_state="None" key_name="None">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" type="fixed" addr="192.168.0.3" mac_addr=
"aa:bb:cc:dd:ee:ff"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v3/servers/348a9852-69c2-4dc2-
a78d-fe7af4c2bf19" rel="self"/>
    <atom:link href="http://openstack.example.com/servers/348a9852-69c2-4dc2-
a78d-fe7af4c2bf19" rel="bookmark"/>
</server>
```

This operation does not return a response body.

## 5.12.2. List servers with details for user

Method	URI	Description
GET	/v3/servers/detail	Lists servers with details for a specified user.

**Normal response codes:** 200

### 5.12.2.1. Response

#### Example 5.99. List servers with details for user: JSON response

```
{
    "servers": [
        {
            "addresses": {
                "private": [
                    {
                        "addr": "192.168.0.3",
                        "mac_addr": "aa:bb:cc:dd:ee:ff",
                        "type": "fixed",
                        "version": 4
                    }
                ]
            },
            "created": "2013-09-16T03:07:09Z",
            "flavor": {
                "id": "1",
                "links": [
                    {
                        "href": "http://openstack.example.com/flavors/1",
                        "rel": "bookmark"
                    }
                ]
            },
            "host_id": "a275e77473e464558c4aba0d68e1914d1164e7ee2f69affde7aaae2b",
            "id": "6c8b5385-e74c-4fd5-add6-2fcf42d74a98",
            "image": {
                "id": "70a599e0-31e7-49b7-b260-868f441e862b",
                "links": [
                    {
                        "href": "http://glance.openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
                        "rel": "bookmark"
                    }
                ]
            },
            "key_name": null,
            "links": [
                {
                    "href": "http://openstack.example.com/v3/servers/6c8b5385-e74c-4fd5-add6-2fcf42d74a98",
                    "rel": "self"
                },
                {

```

```

        "href": "http://openstack.example.com/servers/6c8b5385-
e74c-4fd5-add6-2fcf42d74a98",
        "rel": "bookmark"
    },
],
"metadata": {
    "My Server Name": "Apache1"
},
"name": "new-server-test",
"os-extended-status:locked_by": null,
"os-extended-status:power_state": 1,
"os-extended-status:task_state": null,
"os-extended-status:vm_state": "active",
"progress": 0,
"status": "ACTIVE",
"tenant_id": "openstack",
"updated": "2013-09-16T03:07:10Z",
"user_id": "fake"
}
]
}
}

```

### Example 5.100. List servers with details for user: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:os-extended-status="http://docs.openstack.org/compute/ext/
extended_status/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2013-09-16T03:07:14Z"
    user_id="fake" name="new-server-test" created=
    "2013-09-16T03:07:12Z" tenant_id="openstack" progress="0" host_id=
    "37b4264fc07f9cadef91f833e7dbe0123d35f90c9980claf76aa627f" id=
    "712915a0-9acb-4ff5-aa4c-f546cd50d3f6" os-extended-status:vm_state="active"
    os-extended-status:locked_by="None" os-extended-status:power_state="1" os-
    extended-status:task_state="None" key_name="None">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/
>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" type="fixed" addr="192.168.0.3" mac_addr=
                "aa:bb:cc:dd:ee:ff"/>
            </network>
        </addresses>
        <atom:link href="http://openstack.example.com/v3/servers/
712915a0-9acb-4ff5-aa4c-f546cd50d3f6" rel="self"/>
        <atom:link href="http://openstack.example.com/servers/712915a0-9acb-4ff5-
aa4c-f546cd50d3f6" rel="bookmark"/>
    </server>
</servers>

```

This operation does not return a response body.

## 5.13. Flavor access (os-flavor-access)

Flavor access support.

Method	URI	Description
<b>POST</b>	/v3/flavors	Adds access attribute to the flavor create response.
<b>GET</b>	/v3/flavors/detail	Extends flavor detail to add access attribute to the response of flavor detail.
<b>GET</b>	/v3/flavors/1	Extends flavor show to add access attribute to the flavor show response.
<b>POST</b>	/v3/flavors/10/action	Adds flavor access for tenant.
<b>POST</b>	/v3/flavors/10/action	Removes flavor access for tenant.
<b>GET</b>	/v3/flavors/10/os-flavor-access	Returns access list by flavor id.

## 5.13.1. Add access attribute to flavor create

Method	URI	Description
POST	/v3/flavors	Adds access attribute to the flavor create response.

**Normal response codes:** 200

### 5.13.1.1. Request

#### Example 5.101. Add access attribute to flavor create: JSON request

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

#### Example 5.102. Add access attribute to flavor create: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns="http://docs.openstack.org/compute/api/v1.1"
         xmlns:flavor-access="http://docs.openstack.org/compute/core/flavor-access/
api/v3"
         name="test_flavor"
         ram="1024"
         vcpus="2"
         disk="10"
         id="10"
         flavor-access:is_public="False"
/>
```

This operation does not accept a request body.

### 5.13.1.2. Response

#### Example 5.103. Add access attribute to flavor create: JSON response

```
{
  "flavor": {
    "disabled": false,
    "disk": 10,
    "ephemeral": 0,
    "flavor-access:is_public": false,
    "id": "10",
    "links": [
      {
        "href": "http://openstack.example.com/v3/flavors/10",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/v3/flavors/10",
        "rel": "bookmark"
      }
    ]
  }
}
```

```
        "href": "http://openstack.example.com/flavors/10",
        "rel": "bookmark"
    },
    "name": "test_flavor",
    "ram": 1024,
    "swap": 0,
    "vcpus": 2
}
}
```

#### Example 5.104. Add access attribute to flavor create: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:flavor-access="http://docs.openstack.org/compute/core/
flavor-access/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1" name="test_flavor" ram="1024"
ephemeral="0" disabled="False" vcpus="2" swap="0" disk="10" id="10" flavor-
access:is_public="False">
    <atom:link href="http://openstack.example.com/v3/flavors/10" rel="self"/>
    <atom:link href="http://openstack.example.com/flavors/10" rel="bookmark"/>
</flavor>
```

This operation does not return a response body.

## 5.13.2. Add access attribute to flavor detail

Method	URI	Description
GET	/v3/flavors/detail	Extends flavor detail to add access attribute to the response of flavor detail.

Normal response codes: 200

### 5.13.2.1. Response

#### Example 5.105. Add access attribute to flavor detail: JSON response

```
{  
    "flavors": [  
        {  
            "disabled": false,  
            "disk": 1,  
            "ephemeral": 0,  
            "flavor-access:is_public": true,  
            "id": "1",  
            "links": [  
                {  
                    "href": "http://openstack.example.com/v3/flavors/1",  
                    "rel": "self"  
                },  
                {  
                    "href": "http://openstack.example.com/flavors/1",  
                    "rel": "bookmark"  
                }  
            ],  
            "name": "m1.tiny",  
            "ram": 512,  
            "swap": 0,  
            "vcpus": 1  
        },  
        {  
            "disabled": false,  
            "disk": 20,  
            "ephemeral": 0,  
            "flavor-access:is_public": true,  
            "id": "2",  
            "links": [  
                {  
                    "href": "http://openstack.example.com/v3/flavors/2",  
                    "rel": "self"  
                },  
                {  
                    "href": "http://openstack.example.com/flavors/2",  
                    "rel": "bookmark"  
                }  
            ],  
            "name": "m1.small",  
            "ram": 2048,  
            "swap": 0,  
            "vcpus": 1  
        },  
        {  
            "disabled": false,  
            "disk": 40,  
            "ephemeral": 0,  
            "flavor-access:is_public": true,  
            "id": "3",  
            "links": [  
                {  
                    "href": "http://openstack.example.com/v3/flavors/3",  
                    "rel": "self"  
                },  
                {  
                    "href": "http://openstack.example.com/flavors/3",  
                    "rel": "bookmark"  
                }  
            ],  
            "name": "m1.medium",  
            "ram": 4096,  
            "swap": 0,  
            "vcpus": 2  
        }  
    ]  
}
```

```
"disk": 40,
"ephemeral": 0,
"flavor-access:is_public": true,
"id": "3",
"links": [
    {
        "href": "http://openstack.example.com/v3/flavors/3",
        "rel": "self"
    },
    {
        "href": "http://openstack.example.com/flavors/3",
        "rel": "bookmark"
    }
],
"name": "m1.medium",
"ram": 4096,
"swap": 0,
"vcpus": 2
},
{
    "disabled": false,
    "disk": 80,
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "4",
    "links": [
        {
            "href": "http://openstack.example.com/v3/flavors/4",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/flavors/4",
            "rel": "bookmark"
        }
    ],
    "name": "m1.large",
    "ram": 8192,
    "swap": 0,
    "vcpus": 4
},
{
    "disabled": false,
    "disk": 160,
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "5",
    "links": [
        {
            "href": "http://openstack.example.com/v3/flavors/5",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/flavors/5",
            "rel": "bookmark"
        }
    ],
    "name": "m1.xlarge",
    "ram": 16384,
    "swap": 0,
    "vcpus": 8
}
```

```
        }
    ]
}
```

### Example 5.106. Add access attribute to flavor detail: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:flavor-access="http://docs.openstack.org/compute/core/flavor-
access/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
    <flavor name="m1.tiny" ram="512" ephemeral="0" disabled="False" vcpus="1"
swap="0" disk="1" id="1" flavor-access:is_public="True">
        <atom:link href="http://openstack.example.com/v3/flavors/1" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
    </flavor>
    <flavor name="m1.small" ram="2048" ephemeral="0" disabled="False" vcpus="1"
swap="0" disk="20" id="2" flavor-access:is_public="True">
        <atom:link href="http://openstack.example.com/v3/flavors/2" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/2" rel="bookmark"/>
    </flavor>
    <flavor name="m1.medium" ram="4096" ephemeral="0" disabled="False" vcpus="2"
swap="0" disk="40" id="3" flavor-access:is_public="True">
        <atom:link href="http://openstack.example.com/v3/flavors/3" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/3" rel="bookmark"/>
    </flavor>
    <flavor name="m1.large" ram="8192" ephemeral="0" disabled="False" vcpus="4"
swap="0" disk="80" id="4" flavor-access:is_public="True">
        <atom:link href="http://openstack.example.com/v3/flavors/4" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/4" rel="bookmark"/>
    </flavor>
    <flavor name="m1.xlarge" ram="16384" ephemeral="0" disabled="False" vcpus=
"8" swap="0" disk="160" id="5" flavor-access:is_public="True">
        <atom:link href="http://openstack.example.com/v3/flavors/5" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/5" rel="bookmark"/>
    </flavor>
</flavors>
```

This operation does not return a response body.

## 5.13.3. Add access attribute to flavor show

Method	URI	Description
GET	/v3/flavors/1	Extends flavor show to add access attribute to the flavor show response.

**Normal response codes:** 200

### 5.13.3.1. Response

#### Example 5.107. Add access attribute to flavor show: JSON response

```
{
    "flavor": {
        "disabled": false,
        "disk": 1,
        "ephemeral": 0,
        "flavor-access:is_public": true,
        "id": "1",
        "links": [
            {
                "href": "http://openstack.example.com/v3/flavors/1",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/flavors/1",
                "rel": "bookmark"
            }
        ],
        "name": "m1.tiny",
        "ram": 512,
        "swap": 0,
        "vcpus": 1
    }
}
```

#### Example 5.108. Add access attribute to flavor show: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:flavor-access="http://docs.openstack.org/compute/core/flavor-
access/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/compute/api/v1.1" name="m1.tiny" ram="512" ephemeral="0"
disabled="False" vcpus="1" swap="0" disk="1" id="1" flavor-access:is_public=
"True">
    <atom:link href="http://openstack.example.com/v3/flavors/1" rel="self"/>
    <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
</flavor>
```

This operation does not return a response body.

## 5.13.4. Add flavor access

Method	URI	Description
POST	/v3/flavors/10/action	Adds flavor access for tenant.

**Normal response codes:** 200

### 5.13.4.1. Request

#### Example 5.109. Add flavor access: JSON request

```
{
    "add_tenant_access": {
        "tenant_id": "fake_tenant"
    }
}
```

#### Example 5.110. Add flavor access: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<add_tenant_access>
    <tenant_id>fake_tenant</tenant_id>
</add_tenant_access>
```

This operation does not accept a request body.

### 5.13.4.2. Response

#### Example 5.111. Add flavor access: JSON response

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

#### Example 5.112. Add flavor access: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor_access>
    <access tenant_id="openstack" flavor_id="10"/>
    <access tenant_id="fake_tenant" flavor_id="10"/>
</flavor_access>
```

This operation does not return a response body.

## 5.13.5. Remove flavor access

Method	URI	Description
POST	/v3/flavors/10/action	Removes flavor access for tenant.

**Normal response codes:** 200

### 5.13.5.1. Request

#### Example 5.113. Remove flavor access: JSON request

```
{
    "remove_tenant_access": {
        "tenant_id": "fake_tenant"
    }
}
```

#### Example 5.114. Remove flavor access: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<remove_tenant_access>
    <tenant_id>fake_tenant</tenant_id>
</remove_tenant_access>
```

This operation does not accept a request body.

### 5.13.5.2. Response

#### Example 5.115. Remove flavor access: JSON response

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

#### Example 5.116. Remove flavor access: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor_access>
    <access tenant_id="openstack" flavor_id="10"/>
</flavor_access>
```

This operation does not return a response body.

## 5.13.6. Return access list

Method	URI	Description
GET	/v3/flavors/10/os-flavor-access	Returns access list by flavor id.

Normal response codes: 200

### 5.13.6.1. Request

#### Example 5.117. Return access list: JSON request

```
{
  "add_tenant_access": {
    "tenant_id": "fake_tenant"
  }
}
```

#### Example 5.118. Return access list: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<add_tenant_access>
  <tenant_id>fake_tenant</tenant_id>
</add_tenant_access>
```

This operation does not accept a request body.

### 5.13.6.2. Response

#### Example 5.119. Return access list: JSON response

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

#### Example 5.120. Return access list: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor_access>
  <access tenant_id="openstack" flavor_id="10"/>
  <access tenant_id="fake_tenant" flavor_id="10"/>
</flavor_access>
```

This operation does not return a response body.

## 5.14. Flavor extra-specs (flavor-extra-specs)

Lists, creates, deletes, and updates the extra-specs or keys for a flavor.

Method	URI	Description
<b>POST</b>	/v3/flavors/{flavor_id}/flavor-extra-specs	Creates and updates flavor extra specs.
<b>GET</b>	/v3/flavors/{flavor_id}/flavor-extra-specs	Lists extra specs for specified flavor.
<b>GET</b>	/v3/flavors/{flavor_id}/flavor-extra-specs/{flavor_extra_spec_key}	Shows an extra spec for specified flavor by the key.
<b>PUT</b>	/v3/flavors/{flavor_id}/flavor-extra-specs/{flavor_extra_spec_key}	Updates specified extra spec value by the key.

## 5.14.1. Create or update flavor extra specs

Method	URI	Description
POST	/v3/flavors/{flavor_id}/flavor-extra-specs	Creates and updates flavor extra specs.

Normal response codes: 201

### 5.14.1.1. Request

#### Example 5.121. Create or update flavor extra specs: JSON request

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

#### Example 5.122. Create or update flavor extra specs: XML request

```
<?xml version="1.0" encoding="UTF-8" ?>
<extra_specs>
    <key1>value1</key1>
    <key2>value2</key2>
</extra_specs>
```

This operation does not accept a request body.

### 5.14.1.2. Response

#### Example 5.123. Create or update flavor extra specs: JSON response

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

#### Example 5.124. Create or update flavor extra specs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extra_specs>
    <key2>value2</key2>
    <key1>value1</key1>
</extra_specs>
```

This operation does not return a response body.

## 5.14.2. Show flavor extra specs

Method	URI	Description
GET	/v3/flavors/{flavor_id}/flavor-extra-specs	Lists extra specs for specified flavor.

Normal response codes: 200

### 5.14.2.1. Request

#### Example 5.125. Show flavor extra specs: JSON request

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

#### Example 5.126. Show flavor extra specs: XML request

```
<?xml version="1.0" encoding="UTF-8" ?>
<extra_specs>
    <key1>value1</key1>
    <key2>value2</key2>
</extra_specs>
```

This operation does not accept a request body.

### 5.14.2.2. Response

#### Example 5.127. Show flavor extra specs: JSON response

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

#### Example 5.128. Show flavor extra specs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extra_specs>
    <key2>value2</key2>
    <key1>value1</key1>
</extra_specs>
```

This operation does not return a response body.

## 5.14.3. Show flavor extra specs

Method	URI	Description
GET	/v3/flavors/{flavor_id}/flavor-extra-specs/{flavor_extra_spec_key}	Shows an extra spec for specified flavor by the key.

Normal response codes: 200

### 5.14.3.1. Request

#### Example 5.129. Show flavor extra specs: JSON request

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

#### Example 5.130. Show flavor extra specs: XML request

```
<?xml version="1.0" encoding="UTF-8" ?>
<extra_specs>
    <key1>value1</key1>
    <key2>value2</key2>
</extra_specs>
```

This operation does not accept a request body.

### 5.14.3.2. Response

#### Example 5.131. Show flavor extra specs: JSON response

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

#### Example 5.132. Show flavor extra specs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extra_spec key="key1">value1</extra_spec>
```

This operation does not return a response body.

## 5.14.4. Update flavor extra spec

Method	URI	Description
PUT	/v3/flavors/{flavor_id}/flavor-extra-specs/{flavor_extra_spec_key}	Updates specified extra spec value by the key.

Normal response codes: 200

### 5.14.4.1. Request

#### Example 5.133. Update flavor extra spec: JSON request

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

#### Example 5.134. Update flavor extra spec: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<key1>new_value1</key1>
```

This operation does not accept a request body.

### 5.14.4.2. Response

#### Example 5.135. Update flavor extra spec: JSON response

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

#### Example 5.136. Update flavor extra spec: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extra_spec key="key1">new_value1</extra_spec>
```

This operation does not return a response body.

## 5.15. Flavors manage (flavor-manage)

Support for creating and deleting flavor.

Method	URI	Description
POST	/v3/flavors	Creates a flavor.
DELETE	/v3/flavors/{flavor_id}	Deletes a flavor.

## 5.15.1. Create flavor

Method	URI	Description
POST	/v3/flavors	Creates a flavor.

**Normal response codes:** 200

### 5.15.1.1. Request

#### Example 5.137. Create flavor: JSON request

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

#### Example 5.138. Create flavor: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<flavor>
  <name>test_flavor</name>
  <ram>1024</ram>
  <vcpus>2</vcpus>
  <disk>10</disk>
  <id>10</id>
</flavor>
```

This operation does not accept a request body.

### 5.15.1.2. Response

#### Example 5.139. Create flavor: JSON response

```
{
  "flavor": {
    "disabled": false,
    "disk": 10,
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "10",
    "links": [
      {
        "href": "http://openstack.example.com/v3/flavors/10",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/flavors/10",
        "rel": "bookmark"
      }
    ]
  }
}
```

```
        "name": "test_flavor",
        "ram": 1024,
        "swap": 0,
        "vcpus": 2
    }
}
```

### Example 5.140. Create flavor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:flavor-access="http://docs.openstack.org/compute/core/
flavor-access/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1" name="test_flavor" ram="1024"
ephemeral="0" disabled="False" vcpus="2" swap="0" disk="10" id="10" flavor-
access:is_public="True">
    <atom:link href="http://openstack.example.com/v3/flavors/10" rel="self"/>
    <atom:link href="http://openstack.example.com/flavors/10" rel="bookmark"/>
</flavor>
```

This operation does not return a response body.

## 5.15.2. Delete flavor

Method	URI	Description
<b>DELETE</b>	/v3/flavors/{flavor_id}	Deletes a flavor.

**Normal response codes:** 204

### 5.15.2.1. Request

#### Example 5.141. Delete flavor: JSON request

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

#### Example 5.142. Delete flavor: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<flavor>
  <name>test_flavor</name>
  <ram>1024</ram>
  <vcpus>2</vcpus>
  <disk>10</disk>
  <id>10</id>
</flavor>
```

This operation does not accept a request body.

## 5.16. Flavors with rxtx\_factor extended attribute (os-flavor-rxtx)

Support to show the rxtx status of a flavor.

Method	URI	Description
<b>POST</b>	/v3/flavors	Creates a flavor.
<b>GET</b>	/v3/flavors/detail	Lists flavors with details.
<b>GET</b>	/v3/flavors/{flavor_id}	Shows details for a specified flavor.

## 5.16.1. Create flavor

Method	URI	Description
POST	/v3/flavors	Creates a flavor.

**Normal response codes:** 200

### 5.16.1.1. Request

#### Example 5.143. Create flavor: JSON request

```
{
  "flavor": {
    "name": "flavortest",
    "ram": 1024,
    "vcpus": 2,
    "disk": 10,
    "id": "100",
    "rxtx_factor": 2.0
  }
}
```

#### Example 5.144. Create flavor: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<flavor xmlns="http://docs.openstack.org/compute/api/v1.1"
         name="flavortest"
         ram="1024"
         vcpus="2"
         disk="10"
         id="100"
         rxtx_factor="2.0" />
```

This operation does not accept a request body.

### 5.16.1.2. Response

#### Example 5.145. Create flavor: JSON response

```
{
  "flavor": {
    "disabled": false,
    "disk": 10,
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "100",
    "links": [
      {
        "href": "http://openstack.example.com/v3/flavors/100",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/flavors/100",
        "rel": "bookmark"
      }
    ]
  }
}
```

```
        ],
        "name": "flavortest",
        "ram": 1024,
        "rxtx_factor": 2.0,
        "swap": 0,
        "vcpus": 2
    }
}
```

### Example 5.146. Create flavor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:flavor-access="http://docs.openstack.org/compute/core/
flavor-access/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1" name="flavortest" ram="1024"
ephemeral="0" disabled="False" vcpus="2" swap="0" disk="10" id="100" flavor-
access:is_public="True" rxtx_factor="2.0">
    <atom:link href="http://openstack.example.com/v3/flavors/100" rel="self"/>
    <atom:link href="http://openstack.example.com/flavors/100" rel="bookmark"/>
</flavor>
```

This operation does not return a response body.

## 5.16.2. List flavors with details

Method	URI	Description
GET	/v3/flavors/detail	Lists flavors with details.

Normal response codes: 200

### 5.16.2.1. Response

#### Example 5.147. List flavors with details: JSON response

```
{
    "flavors": [
        {
            "disabled": false,
            "disk": 1,
            "ephemeral": 0,
            "flavor-access:is_public": true,
            "id": "1",
            "links": [
                {
                    "href": "http://openstack.example.com/v3/flavors/1",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/flavors/1",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.tiny",
            "ram": 512,
            "rxtx_factor": 1.0,
            "swap": 0,
            "vcpus": 1
        },
        {
            "disabled": false,
            "disk": 20,
            "ephemeral": 0,
            "flavor-access:is_public": true,
            "id": "2",
            "links": [
                {
                    "href": "http://openstack.example.com/v3/flavors/2",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/flavors/2",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.small",
            "ram": 2048,
            "rxtx_factor": 1.0,
            "swap": 0,
            "vcpus": 1
        }
    ]
}
```

```
    "disabled": false,
    "disk": 40,
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "3",
    "links": [
        {
            "href": "http://openstack.example.com/v3/flavors/3",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/flavors/3",
            "rel": "bookmark"
        }
    ],
    "name": "m1.medium",
    "ram": 4096,
    "rxtx_factor": 1.0,
    "swap": 0,
    "vcpus": 2
},
{
    "disabled": false,
    "disk": 80,
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "4",
    "links": [
        {
            "href": "http://openstack.example.com/v3/flavors/4",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/flavors/4",
            "rel": "bookmark"
        }
    ],
    "name": "m1.large",
    "ram": 8192,
    "rxtx_factor": 1.0,
    "swap": 0,
    "vcpus": 4
},
{
    "disabled": false,
    "disk": 160,
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "5",
    "links": [
        {
            "href": "http://openstack.example.com/v3/flavors/5",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/flavors/5",
            "rel": "bookmark"
        }
    ],
    "name": "m1.xlarge",
```

```
        "ram": 16384,
        "rxtx_factor": 1.0,
        "swap": 0,
        "vcpus": 8
    }
]
}
```

### Example 5.148. List flavors with details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:flavor-access="http://docs.openstack.org/compute/core/flavor-
access/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
    <flavor name="m1.tiny" ram="512" ephemeral="0" disabled="False" vcpus="1"
swap="0" disk="1" id="1" flavor-access:is_public="True" rxtx_factor="1.0">
        <atom:link href="http://openstack.example.com/v3/flavors/1" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
    </flavor>
    <flavor name="m1.small" ram="2048" ephemeral="0" disabled="False" vcpus="1"
swap="0" disk="20" id="2" flavor-access:is_public="True" rxtx_factor="1.0">
        <atom:link href="http://openstack.example.com/v3/flavors/2" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/2" rel="bookmark"/>
    </flavor>
    <flavor name="m1.medium" ram="4096" ephemeral="0" disabled="False" vcpus="2"
swap="0" disk="40" id="3" flavor-access:is_public="True" rxtx_factor="1.0">
        <atom:link href="http://openstack.example.com/v3/flavors/3" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/3" rel="bookmark"/>
    </flavor>
    <flavor name="m1.large" ram="8192" ephemeral="0" disabled="False" vcpus="4"
swap="0" disk="80" id="4" flavor-access:is_public="True" rxtx_factor="1.0">
        <atom:link href="http://openstack.example.com/v3/flavors/4" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/4" rel="bookmark"/>
    </flavor>
    <flavor name="m1.xlarge" ram="16384" ephemeral="0" disabled="False" vcpus=
"8" swap="0" disk="160" id="5" flavor-access:is_public="True" rxtx_factor="1.
0">
        <atom:link href="http://openstack.example.com/v3/flavors/5" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/5" rel="bookmark"/>
    </flavor>
</flavors>
```

This operation does not return a response body.

## 5.16.3. Show flavor details

Method	URI	Description
GET	/v3/flavors/{flavor_id}	Shows details for a specified flavor.

**Normal response codes:** 200

### 5.16.3.1. Response

#### Example 5.149. Show flavor details: JSON response

```
{
  "flavor": {
    "disabled": false,
    "disk": 1,
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "1",
    "links": [
      {
        "href": "http://openstack.example.com/v3/flavors/1",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/flavors/1",
        "rel": "bookmark"
      }
    ],
    "name": "m1.tiny",
    "ram": 512,
    "rxtx_factor": 1.0,
    "swap": 0,
    "vcpus": 1
  }
}
```

#### Example 5.150. Show flavor details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:flavor-access="http://docs.openstack.org/compute/core/flavor-access/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" name="m1.tiny" ram="512" ephemeral="0" disabled="False" vcpus="1" swap="0" disk="1" id="1" flavor-access:is_public="True" rxtx_factor="1.0">
  <atom:link href="http://openstack.example.com/v3/flavors/1" rel="self"/>
  <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
</flavor>
```

This operation does not return a response body.

## 5.17. Flavors with extended attributes (flavors)

Returns information about Flavors.

Method	URI	Description
GET	/v3/flavors	Lists flavors.

Method	URI	Description
<b>GET</b>	/v3/flavors/{flavor_id}	Shows information for a specified flavor.
<b>GET</b>	/v3/flavors/detail	Lists flavors with details.

## 5.17.1. List flavors

Method	URI	Description
GET	/v3/flavors	Lists flavors.

**Normal response codes:** 200

### 5.17.1.1. Request

#### Example 5.151. List flavors: JSON request

```
{
  "flavors": [
    {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/v3/flavors/1",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/flavors/1",
          "rel": "bookmark"
        }
      ],
      "name": "m1.tiny"
    },
    {
      "id": "2",
      "links": [
        {
          "href": "http://openstack.example.com/v3/flavors/2",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/flavors/2",
          "rel": "bookmark"
        }
      ],
      "name": "m1.small"
    },
    {
      "id": "3",
      "links": [
        {
          "href": "http://openstack.example.com/v3/flavors/3",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/flavors/3",
          "rel": "bookmark"
        }
      ],
      "name": "m1.medium"
    },
    {
      "id": "4",
      "links": [
        {
          "href": "http://openstack.example.com/v3/flavors/4",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/flavors/4",
          "rel": "bookmark"
        }
      ],
      "name": "m1.large"
    }
  ]
}
```

```
        "links": [
            {
                "href": "http://openstack.example.com/v3/flavors/4",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/flavors/4",
                "rel": "bookmark"
            }
        ],
        "name": "m1.large"
    },
    {
        "id": "5",
        "links": [
            {
                "href": "http://openstack.example.com/v3/flavors/5",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/flavors/5",
                "rel": "bookmark"
            }
        ],
        "name": "m1.xlarge"
    }
]
```

### Example 5.152. List flavors: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
    <flavor name="m1.tiny" id="1">
        <atom:link href="http://openstack.example.com/v3/flavors/1" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
    </flavor>
    <flavor name="m1.small" id="2">
        <atom:link href="http://openstack.example.com/v3/flavors/2" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/2" rel="bookmark"/>
    </flavor>
    <flavor name="m1.medium" id="3">
        <atom:link href="http://openstack.example.com/v3/flavors/3" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/3" rel="bookmark"/>
    </flavor>
    <flavor name="m1.large" id="4">
        <atom:link href="http://openstack.example.com/v3/flavors/4" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/4" rel="bookmark"/>
    </flavor>
    <flavor name="m1.xlarge" id="5">
        <atom:link href="http://openstack.example.com/v3/flavors/5" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/5" rel="bookmark"/>
    </flavor>
</flavors>
```

This operation does not accept a request body.

## 5.17.2. Show flavor details

Method	URI	Description
GET	/v3/flavors/{flavor_id}	Shows information for a specified flavor.

**Normal response codes:** 200

### 5.17.2.1. Request

#### Example 5.153. Show flavor details: JSON request

```
{
  "flavor": {
    "disabled": false,
    "disk": 1,
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "1",
    "links": [
      {
        "href": "http://openstack.example.com/v3/flavors/1",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/flavors/1",
        "rel": "bookmark"
      }
    ],
    "name": "m1.tiny",
    "ram": 512,
    "swap": 0,
    "vcpus": 1
  }
}
```

#### Example 5.154. Show flavor details: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:flavor-access="http://docs.openstack.org/compute/core/flavor-
access/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/compute/api/v1.1" name="m1.tiny" ram="512" ephemeral="0"
disabled="False" vcpus="1" swap="0" disk="1" id="1" flavor-access:is_public=
"True">
  <atom:link href="http://openstack.example.com/v3/flavors/1" rel="self"/>
  <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
</flavor>
```

This operation does not accept a request body.

## 5.17.3. List detailed flavors

Method	URI	Description
GET	/v3/flavors/detail	Lists flavors with details.

**Normal response codes:** 200

### 5.17.3.1. Request

#### Example 5.155. List detailed flavors: JSON request

```
{
  "flavors": [
    {
      "disabled": false,
      "disk": 1,
      "ephemeral": 0,
      "flavor-access:is_public": true,
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/v3/flavors/1",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/flavors/1",
          "rel": "bookmark"
        }
      ],
      "name": "m1.tiny",
      "ram": 512,
      "swap": 0,
      "vcpus": 1
    },
    {
      "disabled": false,
      "disk": 20,
      "ephemeral": 0,
      "flavor-access:is_public": true,
      "id": "2",
      "links": [
        {
          "href": "http://openstack.example.com/v3/flavors/2",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/flavors/2",
          "rel": "bookmark"
        }
      ],
      "name": "m1.small",
      "ram": 2048,
      "swap": 0,
      "vcpus": 1
    },
    {
      "disabled": false,
      "disk": 40,
```

```
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "3",
    "links": [
        {
            "href": "http://openstack.example.com/v3/flavors/3",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/flavors/3",
            "rel": "bookmark"
        }
    ],
    "name": "m1.medium",
    "ram": 4096,
    "swap": 0,
    "vcpus": 2
},
{
    "disabled": false,
    "disk": 80,
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "4",
    "links": [
        {
            "href": "http://openstack.example.com/v3/flavors/4",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/flavors/4",
            "rel": "bookmark"
        }
    ],
    "name": "m1.large",
    "ram": 8192,
    "swap": 0,
    "vcpus": 4
},
{
    "disabled": false,
    "disk": 160,
    "ephemeral": 0,
    "flavor-access:is_public": true,
    "id": "5",
    "links": [
        {
            "href": "http://openstack.example.com/v3/flavors/5",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/flavors/5",
            "rel": "bookmark"
        }
    ],
    "name": "m1.xlarge",
    "ram": 16384,
    "swap": 0,
    "vcpus": 8
}
```

```
    ]
}
```

### Example 5.156. List detailed flavors: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:flavor-access="http://docs.openstack.org/compute/core/flavor-
access/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
    <flavor name="m1.tiny" ram="512" ephemeral="0" disabled="False" vcpus="1"
swap="0" disk="1" id="1" flavor-access:is_public="True">
        <atom:link href="http://openstack.example.com/v3/flavors/1" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
    </flavor>
    <flavor name="m1.small" ram="2048" ephemeral="0" disabled="False" vcpus="1"
swap="0" disk="20" id="2" flavor-access:is_public="True">
        <atom:link href="http://openstack.example.com/v3/flavors/2" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/2" rel="bookmark"/>
    </flavor>
    <flavor name="m1.medium" ram="4096" ephemeral="0" disabled="False" vcpus="2"
swap="0" disk="40" id="3" flavor-access:is_public="True">
        <atom:link href="http://openstack.example.com/v3/flavors/3" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/3" rel="bookmark"/>
    </flavor>
    <flavor name="m1.large" ram="8192" ephemeral="0" disabled="False" vcpus="4"
swap="0" disk="80" id="4" flavor-access:is_public="True">
        <atom:link href="http://openstack.example.com/v3/flavors/4" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/4" rel="bookmark"/>
    </flavor>
    <flavor name="m1.xlarge" ram="16384" ephemeral="0" disabled="False" vcpus=
"8" swap="0" disk="160" id="5" flavor-access:is_public="True">
        <atom:link href="http://openstack.example.com/v3/flavors/5" rel="self"/>
        <atom:link href="http://openstack.example.com/flavors/5" rel="bookmark"/>
    </flavor>
</flavors>
```

This operation does not accept a request body.

## 5.18. Hosts (os-hosts)

Manages physical hosts.

Method	URI	Description
<b>GET</b>	/v3/os-hosts	Lists hosts.
<b>PUT</b>	/v3/os-hosts/{host_name}	Enables or puts a host in maintenance mode.
<b>GET</b>	/v3/os-hosts/{host_name}	Shows details for a specified host.
<b>GET</b>	/v3/os-hosts/{host_name}/reboot	Reboots a host.
<b>GET</b>	/v3/os-hosts/{host_name}/shutdown	Shuts down a host.
<b>GET</b>	/v3/os-hosts/{host_name}/startup	Starts a host.

## 5.18.1. List hosts

Method	URI	Description
GET	/v3/os-hosts	Lists hosts.

**Normal response codes:** 200

### 5.18.1.1. Response

#### Example 5.157. 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": "abffdःda96592c4eacf4111c28fddee17",
      "service": "scheduler",
      "zone": "internal"
    },
    {
      "host_name": "a8820f04962a4b4ba9fe2e9540c24094",
      "service": "cells",
      "zone": "internal"
    }
  ]
}
```

#### Example 5.158. List hosts: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hosts>
  <host zone="internal" host_name="dbd50e387bcf4ad4a7cb82c0d4cd1a29" service="conductor"/>
```

```
<host zone="nova" host_name="548b88fe0ad1461b8f2d297ffb6ebb90" service="compute"/>
<host zone="internal" host_name="5c03090b25a04e46be1b77b6bcc515e" service="cert"/>
<host zone="internal" host_name="d354e5c745424dd5829d2af58d5647fc" service="consoleauth"/>
<host zone="internal" host_name="15136bf1ba7040e691cb48ea72ec9fef" service="network"/>
<host zone="internal" host_name="0577e9f8391c436b9ea55226738c8e66" service="scheduler"/>
<host zone="internal" host_name="8d07b10240d14df7b7bf5cdbf178f969" service="cells"/>
</hosts>
```

This operation does not return a response body.

## 5.18.2. Enables host

Method	URI	Description
PUT	/v3/os-hosts/{host_name}	Enables or puts a host in maintenance mode.

**Normal response codes:** 200

### 5.18.2.1. Request

#### Example 5.159. Enables host: JSON request

```
{
  "host": {
    "status": "enable",
    "maintenance_mode": "disable"
  }
}
```

#### Example 5.160. Enables host: XML request

```
<?xml version="1.0" encoding="UTF-8" ?>
<host>
  <status>enable</status>
  <maintenance_mode>disable</maintenance_mode>
</host>
```

This operation does not accept a request body.

### 5.18.2.2. Response

#### Example 5.161. Enables host: JSON response

```
{
  "host": {
    "host": "65c5d5b7e3bd44308e67fc50f362aee6",
    "maintenance_mode": "off_maintenance",
    "status": "enabled"
  }
}
```

#### Example 5.162. Enables host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host>
  <status>enabled</status>
  <maintenance_mode>off_maintenance</maintenance_mode>
  <host>966158fd3a084a7791a11c592d1a5fcb</host>
</host>
```

This operation does not return a response body.

## 5.18.3. Show host details

Method	URI	Description
GET	/v3/os-hosts/{host_name}	Shows details for a specified host.

**Normal response codes:** 200

### 5.18.3.1. Response

#### Example 5.163. 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)"
      }
    }
  ]
}
```

#### Example 5.164. Show host details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host>
  <resource>
    <project>(total)</project>
    <memory_mb>8192</memory_mb>
    <host>ee7466478092405c8f8e001b9ca0afb0</host>
    <cpu>1</cpu>
    <disk_gb>1028</disk_gb>
  </resource>
  <resource>
    <project>(used_now)</project>
```

```
<memory_mb>512</memory_mb>
<host>ee7466478092405c8f8e001b9ca0afb0</host>
<cpu>0</cpu>
<disk_gb>0</disk_gb>
</resource>
<resource>
<project>(used_max)</project>
<memory_mb>0</memory_mb>
<host>ee7466478092405c8f8e001b9ca0afb0</host>
<cpu>0</cpu>
<disk_gb>0</disk_gb>
</resource>
</host>
```

This operation does not return a response body.

## 5.18.4. Reboot host

Method	URI	Description
GET	/v3/os-hosts/{host_name}/reboot	Reboots a host.

**Normal response codes:** 200

### 5.18.4.1. Response

#### Example 5.165. Reboot host: JSON response

```
{  
    "host": {  
        "host": "9557750dbc464741a89c907921c1cb31",  
        "power_action": "reboot"  
    }  
}
```

#### Example 5.166. Reboot host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<host>  
    <host>34e0a2409dab4052b17b4172a2a44172</host>  
    <power_action>reboot</power_action>  
</host>
```

This operation does not return a response body.

## 5.18.5. Shut down host

Method	URI	Description
GET	/v3/os-hosts/{host_name}/shutdown	Shuts down a host.

**Normal response codes:** 200

### 5.18.5.1. Response

#### Example 5.167. Shut down host: JSON response

```
{  
    "host": {  
        "host": "77cfa0002e4d45fe97f185968111b27b",  
        "power_action": "shutdown"  
    }  
}
```

#### Example 5.168. Shut down host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<host>  
    <host>9c1d73144d4d43e28fe32840e35fec91</host>  
    <power_action>shutdown</power_action>  
</host>
```

This operation does not return a response body.

## 5.18.6. Start host

Method	URI	Description
GET	/v3/os-hosts/{host_name}/startup	Starts a host.

**Normal response codes:** 200

### 5.18.6.1. Response

#### Example 5.169. Start host: JSON response

```
{
    "host": {
        "host": "4b392b27930343bbaa27fd5d8328a564",
        "power_action": "startup"
    }
}
```

#### Example 5.170. Start host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host>
    <host>f6bf5a52bd0645cdb7e1b28a1d647f3c</host>
    <power_action>startup</power_action>
</host>
```

This operation does not return a response body.

## 5.19. Hypervisors (os-hypervisors)

Displays extra statistical information from the machine that hosts the hypervisor through the API for the hypervisor (XenAPI or KVM/libvirt).

Method	URI	Description
GET	/v3/os-hypervisors	Lists hypervisors.
GET	/v3/os-hypervisors/statistics	Shows statistics for all hypervisors.
GET	/v3/os-hypervisors/search{?query}	Searches hypervisors by specified host name.
GET	/v3/os-hypervisors/{hypervisor_id}	Shows details for a specified hypervisor.
GET	/v3/os-hypervisors/{hypervisor_id}/uptime	Shows the uptime for a specified hypervisor.
GET	/v3/os-hypervisors/{hypervisor_id}/servers	Lists servers that run on a specified hypervisor.

## 5.19.1. List hypervisors

Method	URI	Description
GET	/v3/os-hypervisors	Lists hypervisors.

**Normal response codes:** 200

### 5.19.1.1. Response

#### Example 5.171. List hypervisors: JSON response

```
{  
    "hypervisors": [  
        {  
            "hypervisor_hostname": "fake-mini",  
            "id": 1  
        }  
    ]  
}
```

#### Example 5.172. List hypervisors: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<hypervisors>  
    <hypervisor id="1" hypervisor_hostname="fake-mini"/>  
</hypervisors>
```

This operation does not return a response body.

## 5.19.2. Show hypervisor statistics

Method	URI	Description
GET	/v3/os-hypervisors/statistics	Shows statistics for all hypervisors.

**Normal response codes:** 200

### 5.19.2.1. Response

#### Example 5.173. Show hypervisor statistics: JSON response

```
{  
    "hypervisor_statistics": {  
        "count": 1,  
        "current_workload": 0,  
        "disk_available_least": 0,  
        "free_disk_gb": 1028,  
        "free_ram_mb": 7680,  
        "local_gb": 1028,  
        "local_gb_used": 0,  
        "memory_mb": 8192,  
        "memory_mb_used": 512,  
        "running_vms": 0,  
        "vcpus": 1,  
        "vcpus_used": 0  
    }  
}
```

#### Example 5.174. Show hypervisor statistics: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<hypervisor_statistics count="1" vcpus_used="0" local_gb_used="0" memory_mb=  
"8192" current_workload="0" vcpus="1" running_vms="0" free_disk_gb="1028"  
disk_available_least="0" local_gb="1028" free_ram_mb="7680" memory_mb_used=  
"512"/>
```

This operation does not return a response body.

## 5.19.3. Search hypervisors

Method	URI	Description
GET	/v3/os-hypervisors/search{?query}	Searches hypervisors by specified host name.

**Normal response codes:** 200

### 5.19.3.1. Response

#### Example 5.175. Search hypervisors: JSON response

```
{  
    "hypervisors": [  
        {  
            "hypervisor_hostname": "fake-mini",  
            "id": 1  
        }  
    ]  
}
```

#### Example 5.176. Search hypervisors: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<hypervisors>  
    <hypervisor id="1" hypervisor_hostname="fake-mini"/>  
</hypervisors>
```

This operation does not return a response body.

## 5.19.4. Show hypervisor details

Method	URI	Description
GET	/v3/os-hypervisors/{hypervisor_id}	Shows details for a specified hypervisor.

**Normal response codes:** 200

### 5.19.4.1. Response

#### Example 5.177. Show hypervisor details: JSON response

```
{
  "hypervisor": {
    "cpu_info": "?",
    "current_workload": 0,
    "disk_available_least": 0,
    "free_disk_gb": 1028,
    "free_ram_mb": 7680,
    "hypervisor_hostname": "fake-mini",
    "hypervisor_type": "fake",
    "hypervisor_version": 1,
    "id": 1,
    "local_gb": 1028,
    "local_gb_used": 0,
    "memory_mb": 8192,
    "memory_mb_used": 512,
    "running_vms": 0,
    "service": {
      "host": "043b3cacf6f34c90a7245151fc8ebcda",
      "id": 2
    },
    "vcpus": 1,
    "vcpus_used": 0
  }
}
```

#### Example 5.178. Show hypervisor details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisor vcpus_used="0" hypervisor_type="fake" local_gb_used="0"
hypervisor_hostname="fake-mini" memory_mb_used="512" memory_mb="8192"
current_workload="0" vcpus="1" cpu_info "?" running_vms="0" free_disk_gb=
"1028" hypervisor_version="1" disk_available_least="0" local_gb="1028"
free_ram_mb="7680" id="1">
  <service host="8151475cf9a44809841b60986c2f2d8e" id="2"/>
</hypervisor>
```

This operation does not return a response body.

## 5.19.5. Show hypervisor uptime

Method	URI	Description
GET	/v3/os-hypervisors/{hypervisor_id}/uptime	Shows the uptime for a specified hypervisor.

**Normal response codes:** 200

### 5.19.5.1. Response

#### Example 5.179. Show hypervisor uptime: JSON response

```
{  
    "hypervisor": {  
        "hypervisor_hostname": "fake-mini",  
        "id": 1,  
        "uptime": " 08:32:11 up 93 days, 18:25, 12 users,  load average: 0.20,  
0.12, 0.14"  
    }  
}
```

#### Example 5.180. Show hypervisor uptime: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<hypervisor uptime=" 08:32:11 up 93 days, 18:25, 12 users,  load average: 0.  
20, 0.12, 0.14" id="1" hypervisor_hostname="fake-mini"/>
```

This operation does not return a response body.

## 5.19.6. List servers for a hypervisor

Method	URI	Description
GET	/v3/os-hypervisors/{hypervisor_id}/servers	Lists servers that run on a specified hypervisor.

Normal response codes: 200

### 5.19.6.1. Request

#### Example 5.181. List servers for a hypervisor: JSON request

```
{
  "hypervisor": {
    "hypervisor_hostname": "fake-mini",
    "id": 1,
    "servers": []
  }
}
```

#### Example 5.182. List servers for a hypervisor: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisor id="1" hypervisor_hostname="fake-mini">
  <servers/>
</hypervisor>
```

This operation does not accept a request body.

### 5.19.6.2. Response

#### Example 5.183. List servers for a hypervisor: JSON response

```
{
  "hypervisors": [
    {
      "hypervisor_hostname": "fake-mini",
      "id": 1
    }
  ]
}
```

#### Example 5.184. List servers for a hypervisor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisors>
  <hypervisor id="1" hypervisor_hostname="fake-mini"/>
</hypervisors>
```

This operation does not return a response body.

## 5.20. Server actions (servers)

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

Method	URI	Description
<b>GET</b>	/v3/servers/{server_id}/os-instance-actions	Lists actions for a specified instance.
<b>GET</b>	/v3/servers/{server_id}/os-instance-actions/{request_id}	Shows information about a specified instance action.

## 5.20.1. List actions for an instance

Method	URI	Description
GET	/v3/servers/{server_id}/os-instance-actions	Lists actions for a specified instance.

**Normal response codes:** 200

### 5.20.1.1. Response

#### Example 5.185. List actions for an instance: JSON response

```
{
  "instance_actions": [
    {
      "action": "resize",
      "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
      "message": "",
      "project_id": "842",
      "request_id": "req-25517360-b757-47d3-be45-0e8d2a01b36a",
      "start_time": "2012-12-05T01:00:00.000000",
      "user_id": "789"
    },
    {
      "action": "reboot",
      "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
      "message": "",
      "project_id": "147",
      "request_id": "req-3293a3f1-b44c-4609-b8d2-d81b105636b8",
      "start_time": "2012-12-05T00:00:00.000000",
      "user_id": "789"
    }
  ]
}
```

#### Example 5.186. List actions for an instance: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<instance_actions>
  <instance_action instance_uuid="b48316c5-71e8-45e4-9884-6c78055b9b13"
    user_id="789" start_time="2012-12-05 01:00:00+00:00" request_id=
    "req-25517360-b757-47d3-be45-0e8d2a01b36a" action="resize" message=""
    project_id="842"/>
  <instance_action instance_uuid="b48316c5-71e8-45e4-9884-6c78055b9b13"
    user_id="789" start_time="2012-12-05 00:00:00+00:00" request_id=
    "req-3293a3f1-b44c-4609-b8d2-d81b105636b8" action="reboot" message=""
    project_id="147"/>
</instance_actions>
```

This operation does not return a response body.

## 5.20.2. Shows instance action

Method	URI	Description
GET	/v3/servers/{server_id}/os-instance-actions/{request_id}	Shows information about a specified instance action.

**Normal response codes:** 200

### 5.20.2.1. Response

#### Example 5.187. Shows instance action: JSON response

```
{
  "instance_action": {
    "action": "reboot",
    "events": [
      {
        "event": "schedule",
        "finish_time": "2012-12-05T01:02:00.000000",
        "result": "Success",
        "start_time": "2012-12-05T01:00:02.000000",
        "traceback": ""
      },
      {
        "event": "compute_create",
        "finish_time": "2012-12-05T01:04:00.000000",
        "result": "Success",
        "start_time": "2012-12-05T01:03:00.000000",
        "traceback": ""
      }
    ],
    "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
    "message": "",
    "project_id": "147",
    "request_id": "req-3293a3f1-b44c-4609-b8d2-d81b105636b8",
    "start_time": "2012-12-05T00:00:00.000000",
    "user_id": "789"
  }
}
```

#### Example 5.188. Shows instance action: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<instance_action instance_uuid="b48316c5-71e8-45e4-9884-6c78055b9b13" user_id="789" start_time="2012-12-05 00:00:00+00:00" request_id="req-3293a3f1-b44c-4609-b8d2-d81b105636b8" action="reboot" message="" project_id="147">
  <events finish_time="2012-12-05 01:02:00+00:00" start_time="2012-12-05 01:00:02+00:00" traceback="" event="schedule" result="Success"/>
  <events finish_time="2012-12-05 01:04:00+00:00" start_time="2012-12-05 01:03:00+00:00" traceback="" event="compute_create" result="Success"/>
</instance_action>
```

This operation does not return a response body.

## 5.21. Instance usage audit log (os-instance-usage-audit-log)

Admin-only task log monitoring.

Method	URI	Description
<b>GET</b>	/v3/os-instance-usage-audit-log	Lists usage audits for a specified instance.
<b>GET</b>	/v3/os-instance-usage-audit-log/{datetime}	Lists instance usage audits that occurred before a specified time.

## 5.21.1. List usage audits

Method	URI	Description
GET	/v3/os-instance-usage-audit-log	Lists usage audits for a specified instance.

**Normal response codes:** 200

### 5.21.1.1. Request

This operation does not accept a request body.

### 5.21.1.2. Response

#### Example 5.189. List usage audits: JSON response

```
{
  "instance_usage_audit_log": {
    "hosts_not_run": [
      "defac351f91940668301096238d26b47"
    ],
    "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": "2013-08-01 00:00:00",
    "period Ending": "2013-09-01 00:00:00",
    "total_errors": 0,
    "total_instances": 0
  }
}
```

#### Example 5.190. List usage audits: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<instance_usage_audit_log>
  <total_errors>0</total_errors>
  <total_instances>0</total_instances>
  <log/>
  <num_hosts_running>0</num_hosts_running>
  <num_hosts_done>0</num_hosts_done>
  <num_hosts_not_run>1</num_hosts_not_run>
  <hosts_not_run>
    <item>68368b25d745464695a6352a855d8d2e</item>
  </hosts_not_run>
  <overall_status>0 of 1 hosts done. 0 errors.</overall_status>
  <period_endin>2013-09-01 00:00:00</period_endin>
  <period_beginning>2013-08-01 00:00:00</period_beginning>
  <num_hosts>1</num_hosts>
</instance_usage_audit_log>
```

This operation does not return a response body.

## 5.21.2. List instance usage

Method	URI	Description
GET	/v3/os-instance-usage-audit-log/{datetime}	Lists instance usage audits that occurred before a specified time.

**Normal response codes:** 200

### 5.21.2.1. Request

This table shows the URI parameters for the list instance usage request:

Name	Type	Description
{datetime}	String	The date and time stamp.

This operation does not accept a request body.

### 5.21.2.2. Response

#### Example 5.191. List instance usage: JSON response

```
{
  "instance_usage_audit_log": {
    "hosts_not_run": [
      "ccf86bf6d0104fe69ee658eaa31a7469"
    ],
    "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
  }
}
```

#### Example 5.192. List instance usage: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<instance_usage_audit_log>
  <total_errors>0</total_errors>
  <total_instances>0</total_instances>
  <log/>
  <num_hosts_running>0</num_hosts_running>
  <num_hosts_done>0</num_hosts_done>
  <num_hosts_not_run>1</num_hosts_not_run>
  <hosts_not_run>
    <item>744e1407f60c44ad832bd9f4ffe2122f</item>
  </hosts_not_run>
  <overall_status>0 of 1 hosts done. 0 errors.</overall_status>
  <period_endin>2012-07-01 00:00:00</period_endin>
  <period_beginning>2012-06-01 00:00:00</period_beginning>
```

```
<num_hosts>1</num_hosts>
</instance_usage_audit_log>
```

This operation does not return a response body.

## 5.22. Limits (limits)

Provide all global and rate limit information.

Method	URI	Description
<b>GET</b>	/v3/limits	Show global and rate limit information.

## 5.22.1. Show limits

Method	URI	Description
GET	/v3/limits	Show global and rate limit information.

**Normal response codes:** 200

### 5.22.1.1. Request

#### Example 5.193. Show limits: JSON request

```
{
  "limits": [
    "rate": [
      {
        "limit": [
          {
            "next-available": "2013-09-09T13:37:32Z",
            "remaining": 10,
            "unit": "MINUTE",
            "value": 10,
            "verb": "POST"
          },
          {
            "next-available": "2013-09-09T13:37:32Z",
            "remaining": 10,
            "unit": "MINUTE",
            "value": 10,
            "verb": "PUT"
          },
          {
            "next-available": "2013-09-09T13:37:32Z",
            "remaining": 100,
            "unit": "MINUTE",
            "value": 100,
            "verb": "DELETE"
          }
        ],
        "regex": ".*",
        "uri": "*"
      },
      {
        "limit": [
          {
            "next-available": "2013-09-09T13:37:32Z",
            "remaining": 50,
            "unit": "DAY",
            "value": 50,
            "verb": "POST"
          }
        ],
        "regex": "^/servers",
        "uri": "*/servers"
      },
      {
        "limit": [
          {
            "next-available": "2013-09-09T13:37:32Z",
            "remaining": 100,
            "unit": "DAY",
            "value": 100,
            "verb": "PUT"
          }
        ],
        "regex": ".*",
        "uri": "*"
      }
    ]
  ]
}
```

```

        "next-available": "2013-09-09T13:37:32Z",
        "remaining": 3,
        "unit": "MINUTE",
        "value": 3,
        "verb": "GET"
    }
],
"regex": ".*changes_since.*",
"uri": "*changes_since"
}
]
}
}

```

#### Example 5.194. Show limits: XML request

```

<?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>
        <rate regex=".*" uri="*"\>
            <limit next-available="2013-09-09T13:37:32Z" unit="MINUTE" verb="POST" remaining="10" value="10"/>
            <limit next-available="2013-09-09T13:37:32Z" unit="MINUTE" verb="PUT" remaining="10" value="10"/>
            <limit next-available="2013-09-09T13:37:32Z" unit="MINUTE" verb="DELETE" remaining="100" value="100"/>
        </rate>
        <rate regex="^/servers" uri="*/servers"\>
            <limit next-available="2013-09-09T13:37:32Z" unit="DAY" verb="POST" remaining="50" value="50"/>
        </rate>
        <rate regex=".*changes_since.*" uri="*changes_since"\>
            <limit next-available="2013-09-09T13:37:32Z" unit="MINUTE" verb="GET" remaining="3" value="3"/>
        </rate>
    </rates>
</limits>

```

This operation does not accept a request body.

## 5.23. Migrations (os-migrations)

Provide data on migrations.

Method	URI	Description
GET	/v3/os-migrations	Lists in-progress migrations.

## 5.23.1. List migrations

Method	URI	Description
GET	/v3/os-migrations	Lists in-progress migrations.

**Normal response codes:** 200

### 5.23.1.1. Response

#### Example 5.195. 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"
    }
  ]
}
```

#### Example 5.196. List migrations: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<migrations>
  <migration dest_host="1.2.3.4" status="Done" old_instance_type_id="1"
  updated_at="2012-10-29 13:42:02" dest_compute="compute2" created_at=
  "2012-10-29 13:42:02" source_node="node1" instance_uuid="instance_id_123"
  dest_node="node2" id="1234" new_instance_type_id="2" source_compute=
  "compute1"/>
  <migration dest_host="5.6.7.8" status="Done" old_instance_type_id="5"
  updated_at="2013-10-22 13:42:02" dest_compute="compute20" created_at=
  "2013-10-22 13:42:02" source_node="node10" instance_uuid="instance_id_456"
```

```
dest_node="node20" id="5678" new_instance_type_id="6" source_compute=
"compute10" />
</migrations>
```

This operation does not return a response body.

## 5.24. Multinic (os-multinic)

Multiple network support.

Method	URI	Description
<b>POST</b>	/v3/servers	Removes an IP from a specified instance.
<b>POST</b>	/v3/servers/action	Adds an IP to a specified network on an instance.

## 5.24.1. Remove IP from instance

Method	URI	Description
POST	/v3/servers	Removes an IP from a specified instance.

Normal response codes: 202

### 5.24.1.1. Request

#### Example 5.197. Remove IP from instance: JSON request

```
{  
    "remove_fixed_ip": {  
        "address": "10.0.0.4"  
    }  
}
```

#### Example 5.198. Remove IP from instance: XML request

```
<remove_fixed_ip>  
    <address>10.0.0.4</address>  
</remove_fixed_ip>
```

This operation does not accept a request body.

## 5.24.2. Add IP to instance

Method	URI	Description
POST	/v3/servers/action	Adds an IP to a specified network on an instance.

**Normal response codes:** 202

### 5.24.2.1. Request

#### Example 5.199. Add IP to instance: JSON request

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

#### Example 5.200. Add IP to instance: XML request

```
<add_fixed_ip>
    <network_id>1</network_id>
</add_fixed_ip>
```

This operation does not accept a request body.

## 5.25. Quota class (os-quota-class-sets)

Quota classes management support.

Method	URI	Description
GET	/v3/os-quota-class-sets/{class_id}	Shows the quota for a specified class.
PUT	/v3/os-quota-class-sets/{class_id}	Updates quota for a specified class.

## 5.25.1. Show quota

Method	URI	Description
GET	/v3/os-quota-class-sets/{class_id}	Shows the quota for a specified class.

**Normal response codes:** 200

### 5.25.1.1. Response

#### Example 5.201. Show quota: JSON response

```
{  
    "quota_class_set": {  
        "cores": 20,  
        "fixed_ips": -1,  
        "floating_ips": 10,  
        "id": "test_class",  
        "instances": 10,  
        "key_pairs": 100,  
        "metadata_items": 128,  
        "ram": 51200,  
        "security_group_rules": 20,  
        "security_groups": 10  
    }  
}
```

#### Example 5.202. Show quota: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<quota_class_set id="test_class">  
    <cores>20</cores>  
    <fixed_ips>-1</fixed_ips>  
    <floating_ips>10</floating_ips>  
    <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_class_set>
```

This operation does not return a response body.

## 5.25.2. Update quota

Method	URI	Description
PUT	/v3/os-quota-class-sets/{class_id}	Updates quota for a specified class.

**Normal response codes:** 200

### 5.25.2.1. Request

#### Example 5.203. Update quota: JSON request

```
{
  "quota_class_set": {
    "instances": 50,
    "cores": 50,
    "ram": 51200,
    "floating_ips": 10,
    "fixed_ips": -1,
    "metadata_items": 128,
    "security_groups": 10,
    "security_group_rules": 20,
    "key_pairs": 100
  }
}
```

#### Example 5.204. Update quota: XML request

```
<?xml version="1.0" encoding="UTF-8" ?>
<quota_class_set>
  <cores>50</cores>
  <floating_ips>10</floating_ips>
  <fixed_ips>-1</fixed_ips>
  <instances>50</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_class_set>
```

This operation does not accept a request body.

### 5.25.2.2. Response

#### Example 5.205. Update quota: JSON response

```
{
  "quota_class_set": {
    "cores": 50,
    "fixed_ips": -1,
    "floating_ips": 10,
    "id": "test_class",
    "instances": 50,
    "key_pairs": 100,
    "metadata_items": 128,
```

```

        "ram": 51200,
        "security_group_rules": 20,
        "security_groups": 10
    }
}

```

### Example 5.206. Update quota: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<quota_class_set id="test_class">
    <cores>50</cores>
    <fixed_ips>-1</fixed_ips>
    <floating_ips>10</floating_ips>
    <instances>50</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_class_set>

```

This operation does not return a response body.

## 5.26. Quota sets (os-quota-sets)

Permits administrators, depending on policy settings, to view quotas for a tenant and view and update default quotas.

Method	URI	Description
<b>DELETE</b>	/v3/os-quota-sets/{tenant_id}	Deletes a quota for tenant.
<b>GET</b>	/v3/os-quota-sets/{tenant_id}	Shows quotas for tenant.
<b>PUT</b>	/v3/os-quota-sets/{tenant_id}	Force-updates quota for tenant.
<b>PUT</b>	/v3/os-quota-sets/{tenant_id}	Updates quota for tenant.
<b>GET</b>	/v3/os-quota-sets/{tenant_id}/defaults	Shows default quotas for tenant.
<b>PUT</b>	/v3/os-quota-sets/{tenant_id}/{?user_id}	Updates quota for user.
<b>DELETE</b>	/v3/os-quota-sets/{tenant_id}/{?user_id}	Deletes quota for a specified user.

## 5.26.1. Delete quota for tenant

Method	URI	Description
<b>DELETE</b>	/v3/os-quota-sets/{tenant_id}	Deletes a quota for tenant.

**Normal response codes:** 204

### 5.26.1.1. Request

This operation does not accept a request body.

## 5.26.2. Show quotas

Method	URI	Description
GET	/v3/os-quota-sets/{tenant_id}	Shows quotas for tenant.

**Normal response codes:** 200

### 5.26.2.1. Request

This operation does not accept a request body.

### 5.26.2.2. Response

#### Example 5.207. Show quotas: JSON response

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

#### Example 5.208. Show quotas: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
    <cores>20</cores>
    <fixed_ips>-1</fixed_ips>
    <floating_ips>10</floating_ips>
    <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.

## 5.26.3. Force-update quota

Method	URI	Description
PUT	/v3/os-quota-sets/{tenant_id}	Force-updates quota for tenant.

**Normal response codes:** 200

### 5.26.3.1. Request

#### Example 5.209. Force-update quota: JSON request

```
{
    "quota_set": {
        "force": "True",
        "instances": 45
    }
}
```

#### Example 5.210. Force-update quota: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set>
    <force>True</force>
    <instances>45</instances>
</quota_set>
```

This operation does not accept a request body.

### 5.26.3.2. Response

#### Example 5.211. Force-update quota: JSON response

```
{
    "quota_set": {
        "cores": 20,
        "fixed_ips": -1,
        "floating_ips": 10,
        "id": "fake_tenant",
        "instances": 45,
        "key_pairs": 100,
        "metadata_items": 128,
        "ram": 51200,
        "security_group_rules": 20,
        "security_groups": 10
    }
}
```

#### Example 5.212. Force-update quota: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
    <cores>20</cores>
    <fixed_ips>-1</fixed_ips>
    <floating_ips>10</floating_ips>
    <instances>45</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.

## 5.26.4. Update quota

Method	URI	Description
PUT	/v3/os-quota-sets/{tenant_id}	Updates quota for tenant.

**Normal response codes:** 200

### 5.26.4.1. Request

#### Example 5.213. Update quota: JSON request

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

#### Example 5.214. Update quota: XML request

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

This operation does not accept a request body.

### 5.26.4.2. Response

#### Example 5.215. Update quota: JSON response

```
{
    "quota_set": {
        "cores": 20,
        "fixed_ips": -1,
        "floating_ips": 10,
        "id": "fake_tenant",
        "instances": 10,
        "key_pairs": 100,
        "metadata_items": 128,
        "ram": 51200,
        "security_group_rules": 20,
        "security_groups": 45
    }
}
```

#### Example 5.216. Update quota: XML response

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

This operation does not return a response body.

## 5.26.5. Show default quotas

Method	URI	Description
GET	/v3/os-quota-sets/{tenant_id}/defaults	Shows default quotas for tenant.

Normal response codes: 200

### 5.26.5.1. Request

This operation does not accept a request body.

### 5.26.5.2. Response

#### Example 5.217. Show default quotas: JSON response

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

#### Example 5.218. Show default quotas: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<quota_set id="fake_tenant">  
    <cores>20</cores>  
    <fixed_ips>-1</fixed_ips>  
    <floating_ips>10</floating_ips>  
    <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.

## 5.26.6. Update quota for user

Method	URI	Description
PUT	/v3/os-quota-sets/{tenant_id}/{?user_id}	Updates quota for user.

Normal response codes: 200

### 5.26.6.1. Request

#### Example 5.219. Update quota for user: JSON request

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

#### Example 5.220. Update quota for user: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set>
  <force>True</force>
  <instances>9</instances>
</quota_set>
```

This operation does not accept a request body.

### 5.26.6.2. Response

#### Example 5.221. Update quota for user: JSON response

```
{
  "quota_set": {
    "cores": 20,
    "fixed_ips": -1,
    "floating_ips": 10,
    "id": "fake_tenant",
    "instances": 9,
    "key_pairs": 100,
    "metadata_items": 128,
    "ram": 51200,
    "security_group_rules": 20,
    "security_groups": 10
  }
}
```

#### Example 5.222. Update quota for user: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <cores>20</cores>
  <fixed_ips>-1</fixed_ips>
  <floating_ips>10</floating_ips>
```

```
<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.

## 5.26.7. Delete quota for user

Method	URI	Description
<b>DELETE</b>	/v3/os-quota-sets/{tenant_id}/{?user_id}	Deletes quota for a specified user.

Normal response codes: 204

### 5.26.7.1. Request

This operation does not accept a request body.

## 5.27. Server remote console (os-remote-consoles)

Interactive console support.

Method	URI	Description
<b>POST</b>	/v3/servers/{server_id}/actions	Gets text console output.
<b>POST</b>	/v3/servers/{server_id}/actions	Gets text console output for VNC.

## 5.27.1. Get output for spice

Method	URI	Description
POST	/v3/servers/{server_id}/actions	Gets text console output.

**Normal response codes:** 200

### 5.27.1.1. Request

#### Example 5.223. Get output for spice: JSON request

```
{
    "get_spice_console": {
        "type": "spice-html5"
    }
}
```

#### Example 5.224. Get output for spice: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<get_spice_console>
    <type>spice-html5</type>
</get_spice_console>
```

This operation does not accept a request body.

### 5.27.1.2. Response

#### Example 5.225. Get output for spice: JSON response

```
{
    "console": {
        "type": "spice-html5",
        "url": "http://127.0.0.1:6082/spice_auto.html?token=a30e5d08-6a20-4043-958f-0852440c6af4"
    }
}
```

#### Example 5.226. Get output for spice: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<console>
    <url>http://127.0.0.1:6082/spice_auto.html?token=45eb7f39-5157-4864-a443-41ac2d498748</url>
    <type>spice-html5</type>
</console>
```

This operation does not return a response body.

## 5.27.2. Get console output

Method	URI	Description
POST	/v3/servers/{server_id}/actions	Gets text console output for VNC.

Normal response codes: 200

### 5.27.2.1. Request

#### Example 5.227. Get console output: JSON request

```
{
  "get_vnc_console": {
    "type": "novnc"
  }
}
```

#### Example 5.228. Get console output: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<get_vnc_console>
  <type>novnc</type>
</get_vnc_console>
```

This operation does not accept a request body.

### 5.27.2.2. Response

#### Example 5.229. Get console output: JSON response

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

#### Example 5.230. Get console output: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<console>
  <url>http://127.0.0.1:6080/vnc_auto.html?token=ba493af2-559b-4893-951c-a835bdd3f823</url>
  <type>novnc</type>
</console>
```

This operation does not return a response body.

## 5.28. Server usage (os-server-usage)

Adds launched\_at and terminated\_at to servers.

Method	URI	Description
<b>GET</b>	/v3/servers/detail	Lists server details for a specified user.
<b>GET</b>	/v3/servers/{server_id}	Shows details for a specified server.

## 5.28.1. List server details for user

Method	URI	Description
GET	/v3/servers/detail	Lists server details for a specified user.

**Normal response codes:** 200

### 5.28.1.1. Response

#### Example 5.231. List server details for user: JSON response

```
{
  "servers": [
    {
      "addresses": {
        "private": [
          {
            "addr": "192.168.0.3",
            "mac_addr": "aa:bb:cc:dd:ee:ff",
            "type": "fixed",
            "version": 4
          }
        ]
      },
      "created": "2013-08-15T12:04:04Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "host_id": "117535ce0eda7ee02ebffe2c976173629385481ae3f2bded5e14a66b",
      "id": "ae114799-9164-48f5-a036-6ef9310acbc4",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
          {
            "href": "http://glance.openstack.example.com/images/70a599e0-31e7-49b7-b260-868f441e862b",
            "rel": "bookmark"
          }
        ]
      },
      "key_name": null,
      "links": [
        {
          "href": "http://openstack.example.com/v3/servers/ae114799-9164-48f5-a036-6ef9310acbc4",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/servers/ae114799-9164-48f5-a036-6ef9310acbc4",
        }
      ]
    }
  ]
}
```

```
        "rel": "bookmark"
    }
],
"metadata": {
    "My Server Name": "Apache1"
},
"name": "new-server-test",
"os-server-usage:launched_at": "2013-08-15T12:04:05.368766",
"os-server-usage:terminated_at": null,
"progress": 0,
"status": "ACTIVE",
"tenant_id": "openstack",
"updated": "2013-08-15T12:04:05Z",
"user_id": "fake"
}
]
```

### Example 5.232. List server details for user: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:os-server-usage="http://docs.openstack.org/compute/ext/os-
server-usage/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/compute/api/v1.1">
    <server status="ACTIVE" updated="2013-08-15T12:04:07Z"
    user_id="fake" name="new-server-test" created=
    "2013-08-15T12:04:06Z" tenant_id="openstack" progress="0" host_id=
    "897fe6cb84a7ce95220c0f6a3a2b3bbc33e6190f6f166ab32a2e3f59" id=
    "14288a58-1ffc-468d-8412-98fc85650d93" os-server-usage:launched_at="2013-08-15
    12:04:07.293502" os-server-usage:terminated_at="None" key_name="None">
        <image id="70a599e0-31e7-49b7-b260-868f441e862b">
            <atom:link href="http://glance.openstack.example.com/images/
    70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
        </image>
        <flavor id="1">
            <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/
>
        </flavor>
        <metadata>
            <meta key="My Server Name">Apache1</meta>
        </metadata>
        <addresses>
            <network id="private">
                <ip version="4" type="fixed" addr="192.168.0.3" mac_addr=
                "aa:bb:cc:dd:ee:ff"/>
            </network>
        </addresses>
        <atom:link href="http://openstack.example.com/v3/servers/
    14288a58-1ffc-468d-8412-98fc85650d93" rel="self"/>
        <atom:link href="http://openstack.example.com/servers/
    14288a58-1ffc-468d-8412-98fc85650d93" rel="bookmark"/>
    </server>
</servers>
```

This operation does not return a response body.

## 5.28.2. Show server details

Method	URI	Description
GET	/v3/servers/{server_id}	Shows details for a specified server.

Normal response codes: 200

### 5.28.2.1. Request

This operation does not accept a request body.

### 5.28.2.2. Response

#### Example 5.233. Show server details: JSON response

```
{  
    "server": {  
        "addresses": {  
            "private": [  
                {  
                    "addr": "192.168.0.3",  
                    "mac_addr": "aa:bb:cc:dd:ee:ff",  
                    "type": "fixed",  
                    "version": 4  
                }  
            ]  
        },  
        "created": "2013-08-15T08:12:40Z",  
        "flavor": {  
            "id": "1",  
            "links": [  
                {  
                    "href": "http://openstack.example.com/flavors/1",  
                    "rel": "bookmark"  
                }  
            ]  
        },  
        "host_id": "73cf3a40601b63f5992894be2daa3712dd599d1c919984951e21edda",  
        "id": "cee6d136-e378-4cf8-9eec-71797f025991",  
        "image": {  
            "id": "70a599e0-31e7-49b7-b260-868f441e862b",  
            "links": [  
                {  
                    "href": "http://glance.openstack.example.com/images/  
70a599e0-31e7-49b7-b260-868f441e862b",  
                    "rel": "bookmark"  
                }  
            ]  
        },  
        "key_name": null,  
        "links": [  
            {  
                "href": "http://openstack.example.com/v3/servers/cee6d136-  
e378-4cf8-9eec-71797f025991",  
                "rel": "self"  
            }  
        ]  
    }  
}
```

```

        },
        {
            "href": "http://openstack.example.com/servers/cee6d136-
e378-4cfc-9eec-71797f025991",
            "rel": "bookmark"
        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "os-server-usage:launched_at": "2013-08-15T08:12:40.108903",
    "os-server-usage:terminated_at": null,
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2013-08-15T08:12:40Z",
    "user_id": "fake"
}
}

```

### Example 5.234. Show server details: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:os-server-usage="http://docs.openstack.org/compute/ext/
os-server-usage/api/v3" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1" status="ACTIVE" updated=
"2013-08-15T08:12:42Z" user_id="fake" name="new-server-test" created=
"2013-08-15T08:12:42Z" tenant_id="openstack" progress="0" host_id=
"f05316395b5e0b792d2f4f75bc54ef97745900629a11a9cf82a9cdf4" id="63bd45d9-
c0bd-4825-9520-38d3f88225ae" os-server-usage:launched_at="2013-08-15 08:12:42.
949732" os-server-usage:terminated_at="None" key_name="None">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
        <atom:link href="http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
        <atom:link href="http://openstack.example.com/flavors/1" rel="bookmark"/>
    </flavor>
    <metadata>
        <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
        <network id="private">
            <ip version="4" type="fixed" addr="192.168.0.3" mac_addr=
"aa:bb:cc:dd:ee:ff"/>
        </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v3/servers/63bd45d9-
c0bd-4825-9520-38d3f88225ae" rel="self"/>
    <atom:link href="http://openstack.example.com/servers/63bd45d9-
c0bd-4825-9520-38d3f88225ae" rel="bookmark"/>
</server>

```

This operation does not return a response body.

## 5.29. Usage reports (os-simple-tenant-usage)

Provide simple tenant usage for tenant.

Method	URI	Description
<b>GET</b>	/v3/os-simple-tenant-usage	Lists usage information for all tenants.
<b>GET</b>	/v3/os-simple-tenant-usage/{tenant_id}	Shows usage details for a specified tenant.

## 5.29.1. List tenant usage for all tenants

Method	URI	Description
GET	/v3/os-simple-tenant-usage	Lists usage information for all tenants.

**Normal response codes:** 200

### 5.29.1.1. Response

#### Example 5.235. List tenant usage for all tenants: JSON response

```
{
  "tenant_usages": [
    {
      "start": "2013-09-09T13:18:27.898411",
      "stop": "2013-09-09T14:18:27.898411",
      "tenant_id": "openstack",
      "total_hours": 1.0,
      "total_local_gb_usage": 1.0,
      "total_memory_mb_usage": 512.0,
      "total_vcpus_usage": 1.0
    }
  ]
}
```

#### Example 5.236. List tenant usage for all tenants: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<tenant_usages>
  <tenant_usage>
    <tenant_id>openstack</tenant_id>
    <total_local_gb_usage>1.0</total_local_gb_usage>
    <total_vcpus_usage>1.0</total_vcpus_usage>
    <total_memory_mb_usage>512.0</total_memory_mb_usage>
    <total_hours>1.0</total_hours>
    <start>2013-09-09 13:18:27.998271</start>
    <stop>2013-09-09 14:18:27.998271</stop>
    <server_usages/>
  </tenant_usage>
</tenant_usages>
```

This operation does not return a response body.

## 5.29.2. Show usage details for tenant

Method	URI	Description
GET	/v3/os-simple-tenant-us-age/{tenant_id}	Shows usage details for a specified tenant.

**Normal response codes:** 200

### 5.29.2.1. Response

#### Example 5.237. Show usage details for tenant: JSON response

```
{
  "tenant_usage": {
    "server_usages": [
      {
        "ended_at": null,
        "flavor": "m1.tiny",
        "hours": 1.0,
        "instance_id": "bf2fbe78-aelc-4f6b-a68b-390f12b1b983",
        "local_gb": 1,
        "memory_mb": 512,
        "name": "new-server-test",
        "started_at": "2013-09-09T13:18:28.101337",
        "state": "active",
        "tenant_id": "openstack",
        "uptime": 3600,
        "vcpus": 1
      }
    ],
    "start": "2013-09-09T13:18:28.101337",
    "stop": "2013-09-09T14:18:28.101337",
    "tenant_id": "openstack",
    "total_hours": 1.0,
    "total_local_gb_usage": 1.0,
    "total_memory_mb_usage": 512.0,
    "total_vcpus_usage": 1.0
  }
}
```

#### Example 5.238. Show usage details for tenant: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<tenant_usage>
  <tenant_id>openstack</tenant_id>
  <total_local_gb_usage>1.0</total_local_gb_usage>
  <total_vcpus_usage>1.0</total_vcpus_usage>
  <total_memory_mb_usage>512.0</total_memory_mb_usage>
  <total_hours>1.0</total_hours>
  <start>2013-09-09 13:18:27.896728</start>
  <stop>2013-09-09 14:18:27.896728</stop>
  <server_usages>
    <server_usage>
      <instance_id>f14ff0d3-cebe-4692-9202-71946f4db8e4</instance_id>
      <name>new-server-test</name>
      <hours>1.0</hours>
    </server_usage>
  </server_usages>
</tenant_usage>
```

```
<memory_mb>512</memory_mb>
<local_gb>1</local_gb>
<vcpus>1</vcpus>
<tenant_id>openstack</tenant_id>
<flavor>m1.tiny</flavor>
<started_at>2013-09-09 13:18:27.896728</started_at>
<ended_at>None</ended_at>
<state>active</state>
<uptime>3600</uptime>
</server_usage>
</server_usages>
</tenant_usage>
```

This operation does not return a response body.

## 6. Databases Service API v1.0 (CURRENT)

### 6.1. API versions

Method	URI	Description
<b>GET</b>	/	Lists information about all Database Service API versions.
<b>GET</b>	/v1.0	Shows details for the Database Service API v1.0.

## 6.1.1. 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)

### 6.1.1.1. Request

#### Example 6.1. 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.

### 6.1.1.2. Response

#### Example 6.2. 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"
    }
  ]
}
```

## 6.1.2. Show version details

Method	URI	Description
GET	/v1.0	Shows details for the Database Service API v1.0.

**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)

### 6.1.2.1. Request

#### Example 6.3. 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.

### 6.1.2.2. Response

#### Example 6.4. 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"
                }
            ]
        }
    ]
}
```

## 6.2. Database instances (instances)

Method	URI	Description
POST	/v1.0/{accountId}/instances	Creates a database instance.

Method	URI	Description
<b>GET</b>	/v1.0/{accountId}/instances	Lists information, including status, for all database instances.
<b>GET</b>	/v1.0/{accountId}/instances/{instanceId}	Shows database instance details.
<b>DELETE</b>	/v1.0/{accountId}/instances/{instanceId}	Deletes a specified database instance, including any associated data.
<b>POST</b>	/v1.0/{accountId}/instances/{instanceId}/root	Enables the root user for a specified database instance and returns the root password.
<b>GET</b>	/v1.0/{accountId}/instances/{instanceId}/root	Shows root-enabled status for a database instance.

## 6.2.1. Create database instance

Method	URI	Description
POST	/v1.0/{accountId}/instances	Creates a database instance.

This operation asynchronously provisions a new database instance. This call requires the user to specify a flavor and a volume size. The service then provisions the instance with the requested flavor and sets up a volume of the specified size, which is the 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.
- Database instances are directly accessible only on the internal ServiceNet network and using a Cloud resource within the same regional datacenter. For example, a database instance in DFW can only be accessed by a Cloud Server in DFW.

**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)

### 6.2.1.1. 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 specified instance.

### Example 6.5. Create database instance: JSON request

```
POST /v1.0/1234/instances HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: d6cafa5b-e0c7-4ab8-948e-7c95f2acd031
Accept: application/json
Content-Type: application/json
```

```
{
  "instance": {
    "databases": [
      {
        "character_set": "utf8",
        "collate": "utf8_general_ci",
        "name": "sampledb"
      }
    ]
  }
}
```

```

        },
        {
            "name": "nextround"
        }
    ],
    "flavorRef": "https://openstack.example.com/v1.0/1234/flavors/1",
    "name": "json_rack_instance",
    "users": [
        {
            "databases": [
                {
                    "name": "sampledb"
                }
            ],
            "name": "demouser",
            "password": "demopassword"
        }
    ],
    "volume": {
        "size": 2
    }
}
}

```

### 6.2.1.2. Response

#### Example 6.6. 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": "2012-01-25T21:53:10Z",
"volume": {
    "size": 2
}
}
```

For convenience, notice in the response examples above that resources contain links to themselves. This allows a client to easily obtain resource URLs rather than to 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.

## 6.2.2. List database instances

Method	URI	Description
GET	/v1.0/{accountId}/instances	Lists information, including status, for all database instances.

This operation lists the 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)

### 6.2.2.1. 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 specified instance.

#### Example 6.7. 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.

### 6.2.2.2. Response

#### Example 6.8. 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
}
}
]
}
```

## 6.2.3. Show database instance details

Method	URI	Description
GET	/v1.0/{accountId}/instances/{instanceId}	Shows database instance details.

This operation lists the status and details of the specified database instance.

This operation lists the volume size in gigabytes (GB) and the approximate GB used.



### Note

After instance creation, the used size of your volume will be greater than 0. This 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 status for the instance is BUILD, REBOOT, RESIZE, or ERROR.

The list operations return a DNS-resolvable hostname associated with the database instance instead of an IP address. Since the hostname always resolves to the correct IP address of the database instance, this relieves the user from the task of maintaining the mapping. Note that although the IP address may likely change on resizing, migrating, and so forth, the hostname 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)

### 6.2.3.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.

### Example 6.9. 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.

### 6.2.3.2. Response

### Example 6.10. 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
        }
    }
}
```

## 6.2.4. Delete database instance

Method	URI	Description
<b>DELETE</b>	/v1.0/{accountId}/instances/{instanceId}	Deletes a specified database instance, including any associated data.



### Note

This operation does not delete any read slaves.



### Note

This operation is not allowed 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)

### 6.2.4.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.

### Example 6.11. 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.

## 6.2.5. Enable root user

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/root	Enables the root user for a specified 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 you make as a root user may cause detrimental effects to 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. While enabling root does not prevent us from a “best effort” approach to helping you if something goes wrong with your instance, we cannot ensure that we will 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 forth.

**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)

### 6.2.5.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.

### Example 6.12. 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.

### 6.2.5.2. Response

### Example 6.13. 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": "d4311cb1-d912-45be-8517-e8a46f54df66"
    }
}
```

## 6.2.6. 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 specified database instance or false otherwise.

This operation checks an active specified database instance to see if root access is enabled. It returns True if root user is enabled for the specified database instance or False otherwise.

**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)

### 6.2.6.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.

#### Example 6.14. 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.

### 6.2.6.2. Response

#### Example 6.15. 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
}
```

## 6.3. Database instance actions (action)

Method	URI	Description
<b>POST</b>	/v1.0/{accountId}/instances/{instanceId}/action	Restarts the database service on an instance.
<b>POST</b>	/v1.0/{accountId}/instances/{instanceId}/action	Resizes the memory for an instance.
<b>POST</b>	/v1.0/{accountId}/instances/{instanceId}/action	Resizes the volume attached to an instance.

## 6.3.1. Restart instance

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/action	Restarts the database service on an instance.

The restart operation will restart only the MySQL Instance. Restarting MySQL will erase any dynamic configuration settings that you have made within MySQL.



### Note

The MySQL service will be unavailable until the instance restarts.

This operation 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)

### 6.3.1.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.

### Example 6.16. 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": {}
}
```

### 6.3.1.2. Response

### Example 6.17. 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.

## 6.3.2. Resize instance

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/action	Resizes the memory for an instance.

This operation changes the memory size of the instance, assuming a valid flavorRef is provided. Restarts MySQL in the process.

**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)

### 6.3.2.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.

#### Example 6.18. 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"
    }
}
```

### 6.3.2.2. Response

#### Example 6.19. 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.

### 6.3.3. Resize instance volume

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/action	Resizes the volume attached to an instance.

This operation supports resizing the attached volume for an instance. It supports only increasing the volume size and does not support decreasing the size. The volume size is in gigabytes (GB) and must be an integer.



#### Note

You cannot increase the volume to a size larger than the API volume size limit specifies.

This operation 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)

#### 6.3.3.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.

#### Example 6.20. Resize instance volume: JSON request

```
POST /v1.0/1234/instances/23a3d4fb-3731-497b-af4-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
        }
    }
}
```

#### 6.3.3.2. Response

#### Example 6.21. 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.

## 6.4. Databases (databases)

Method	URI	Description
<b>POST</b>	/v1.0/{accountId}/instances/{instanceId}/databases	Creates a database within a specified instance.
<b>GET</b>	/v1.0/{accountId}/instances/{instanceId}/databases	Lists databases for a specified instance.
<b>DELETE</b>	/v1.0/{accountId}/instances/{instanceId}/databases/{databaseName}	Deletes a specified database.

## 6.4.1. Create database

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/databases	Creates a database within a specified instance.

This operation creates a new database within the specified instance.

The name of the database is a required attribute.

The following additional attributes can be specified for each database: `collate` and `character_set`.

**Table 6.1. Required and Optional Attributes for Create 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

See the MySQL documentation for information about supported character sets and collations at <http://dev.mysql.com/doc/refman/5.1/en/charset-mysql.html>.



### Note

The following database names are reserved and cannot be used for creating databases: `lost+found`, `information_schema`, and `mysql`.

See the following tables for information about characters that are valid/invalid for creating database names.

**Table 6.2. Valid Characters That Can Be Used 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 6.3. Characters That Cannot Be Used in a Database Name**

Character
Single quotes
Double quotes
Back quotes
Semicolons
Commas
Backslashes
Forwardslashes

**Table 6.4. 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)

### 6.4.1.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.

#### Example 6.22. 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"
    }
  ]
}
```

### 6.4.1.2. Response

#### Example 6.23. 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.

## 6.4.2. List instance databases

Method	URI	Description
<b>GET</b>	/v1.0/{accountId}/instances/{instanceId}/databases	Lists databases for a specified instance.

This operation lists the databases for the specified 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)

### 6.4.2.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.

#### Example 6.24. 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.

### 6.4.2.2. Response

#### Example 6.25. 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": "exampledb"  
},  
{  
    "name": "nextround"  
},  
{  
    "name": "sampledb"  
},  
{  
    "name": "testingdb"  
}  
]  
}
```

## 6.4.3. Delete database

Method	URI	Description
<b>DELETE</b>	/v1.0/{accountId}/instances/{instanceId}/databases/{databaseName}	Deletes a specified database.

This operation deletes the requested database within the specified database instance. 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)

### 6.4.3.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.
{databaseName}	String	The name for the specified database.

#### Example 6.26. 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.

### 6.4.3.2. Response

#### Example 6.27. 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.

## 6.5. Users (users)

Method	URI	Description
<b>POST</b>	/v1.0/{accountId}/instances/{instanceId}/users	Creates a user for a specified database instance.

Method	URI	Description
<b>GET</b>	/v1.0/{accountId}/instances/{instanceId}/users	Lists the users in a specified database instance.
<b>DELETE</b>	/v1.0/{accountId}/instances/{instanceId}/users/{name}	Deletes a specified user for a specified database instance.

## 6.5.1. Create user

Method	URI	Description
POST	/v1.0/{accountId}/instances/{instanceId}/users	Creates a user for a specified database instance.

This operation asynchronously provisions a new user for the specified database instance based on the configuration defined in the request object. Once the request is validated and progress has started 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 create user. See the request examples for the required xml/json format:

**Table 6.5. Required attributes for create 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. One or more database names must be specified.	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 6.6. Valid Characters That Can Be Used 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 6.7. Characters That Cannot Be Used in a Database Name, User Name, and Password**

Character
Single quotes
Double quotes
Back quotes
Semicolons
Commas
Backslashes
Forwardslashes
Spaces at the front or end of the user name or password

**Table 6.8. 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)

### 6.5.1.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.

#### Example 6.28. 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": "password"
    },
{
    "databases": [
        {
            "name": "databaseB"
        },
        {
            "name": "databaseC"
        }
    ],
    "name": "dbuser4",
    "password": "password"
}
]
```

### 6.5.1.2. Response

#### Example 6.29. 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.

## 6.5.2. List database instance users

Method	URI	Description
GET	/v1.0/{accountId}/instances/{instanceId}/users	Lists the users in a specified database instance.

This operation lists the users in the specified 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 has been 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)

### 6.5.2.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.

### Example 6.30. 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.

### 6.5.2.2. Response

#### Example 6.31. 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"
    }
  ]
}
```

## 6.5.3. Delete user

Method	URI	Description
<b>DELETE</b>	/v1.0/{accountId}/instances/{instanceId}/users/{name}	Deletes a specified user for a specified database instance.

This operation deletes the specified user for the specified database instance.



### Warning

There is a bug in a python library that Rackspace is using that may 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)

### 6.5.3.1. 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 specified instance.
{instanceId}	String	The instance ID for the specified database instance.
{name}	String	The name for the specified user.

### Example 6.32. 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.

### 6.5.3.2. Response

### Example 6.33. 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.

## 6.6. Flavors (flavors)

Method	URI	Description
<b>GET</b>	/v1.0/{accountId}/flavors	Lists information for all available flavors.
<b>GET</b>	/v1.0/{accountId}/flavors/{flavorId}	Shows flavor details.

## 6.6.1. 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)

### 6.6.1.1. 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 specified instance.

#### Example 6.34. 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.

### 6.6.1.2. Response

#### Example 6.35. 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
}
]
```

## 6.6.2. Show flavor details

Method	URI	Description
GET	/v1.0/{accountId}/flavors/{flavorId}	Shows flavor details.

This operation lists all information for the specified flavor ID with details of the RAM.

This resource is identical to the flavors found in the OpenStack Nova 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)

### 6.6.2.1. 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 specified instance.
{flavorId}	String	The flavor ID for the specified flavor.

### Example 6.36. 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.

### 6.6.2.2. Response

### Example 6.37. 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  
    }  
}
```

## 7. Identity API v3 (STABLE)

Get an authentication token that permits access to the OpenStack services REST API.

Method	URI	Description
API versions		
<b>GET</b>	/	Lists information about all Identity API versions.
<b>GET</b>	/v3	Shows details for the Identity API v3.
Tokens		
<b>POST</b>	/v3/auth/tokens	Authenticates and generates a token.
<b>GET</b>	/v3/auth/tokens	Validates a specified token.
<b>HEAD</b>	/v3/auth/tokens	Validates a specified token.
<b>DELETE</b>	/v3/auth/tokens	Revokes a specified token.
Service catalog		
<b>POST</b>	/v3/services	Adds a service.
<b>GET</b>	/v3/services{?type,page,per_page}	Lists services.
<b>GET</b>	/v3/services/{service_id}	Shows details for a specified service.
<b>PATCH</b>	/v3/services/{service_id}	Updates a specified service.
<b>DELETE</b>	/v3/services/{service_id}	Deletes a specified service.
Endpoints		
<b>POST</b>	/v3/endpoints	Adds an endpoint.
<b>GET</b>	/v3/endpoints{?interface,service_id,page,per_page}	Lists available endpoints.
<b>PATCH</b>	/v3/endpoints/{endpoint_id}	Updates a specified endpoint.
<b>DELETE</b>	/v3/endpoints/{endpoint_id}	Deletes a specified endpoint.
Domains		
<b>POST</b>	/v3/domains	Adds a domain.
<b>GET</b>	/v3/domains{?name,enabled,page,per_page}	Lists domains.
<b>GET</b>	/v3/domains/{domain_id}	Shows details for a specified domain.
<b>PATCH</b>	/v3/domains/{domain_id}	Updates a specified domain.
<b>DELETE</b>	/v3/domains/{domain_id}	Deletes a specified domain.
<b>GET</b>	/v3/domains/{domain_id}/users/{user_id}/roles	Lists roles for a user on a domain.
<b>GET</b>	/v3/domains/{domain_id}/groups/{group_id}/roles	Lists roles for a specified domain group.
<b>PUT</b>	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Grants a specified role to a specified domain group.
<b>HEAD</b>	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Validates that a group has a role on a domain.
<b>DELETE</b>	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Revokes a role from a group on a domain.
Projects		
<b>POST</b>	/v3/projects	Adds a project.
<b>GET</b>	/v3/projects{?domain_id,name,enabled,page,per_page}	Lists projects.
<b>GET</b>	/v3/projects/{project_id}	Shows details for a specified project.
<b>PATCH</b>	/v3/projects/{project_id}	Updates a specified project.

Method	URI	Description
<b>DELETE</b>	/v3/projects/{project_id}	Deletes a specified project.
<b>GET</b>	/v3/projects/{project_id}/users/{user_id}/roles	Lists roles for a user in a project.
<b>PUT</b>	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Grants a role to a user on a project.
<b>HEAD</b>	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Validates that a user has a specified role on a project.
<b>DELETE</b>	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Revokes a role from a project user.
<b>GET</b>	/v3/projects/{project_id}/groups/{group_id}/roles	Lists roles for a project group.
<b>PUT</b>	/v3/projects/{project_id}/groups/{group_id}/roles	Grants a role to a project group.
<b>HEAD</b>	/v3/projects/{project_id}/groups/{group_id}/roles	Validates that a project group has a role.
<b>DELETE</b>	/v3/projects/{project_id}/groups/{group_id}/roles	Revokes a role from a project group.
Users		
<b>POST</b>	/v3/users	Adds a user.
<b>GET</b>	/v3/users{?domain_id,email,name(enabled,page,per_page)}	Lists users.
<b>GET</b>	/v3/users/{user_id}	Shows details for a specified user.
<b>PATCH</b>	/v3/users/{user_id}	Updates the password for or enables or disables a specified user.
<b>DELETE</b>	/v3/users/{user_id}	Deletes a specified user.
<b>GET</b>	/v3/users/{user_id}/groups	Lists groups for a specified user.
<b>GET</b>	/v3/users/{user_id}/projects	List projects for a specified user.
<b>GET</b>	/v3/users/{user_id}/roles{?name,project_id, domain_id}	Lists roles for a specified user.
Groups		
<b>POST</b>	/v3/groups	Adds a group.
<b>GET</b>	/v3/groups{?domain_id,page,per_page}	Lists groups.
<b>GET</b>	/v3/groups/{group_id}	Shows details for a specified group.
<b>PATCH</b>	/v3/groups/{group_id}	Updates a specified group.
<b>DELETE</b>	/v3/groups/{group_id}	Deletes a specified group.
<b>GET</b>	/v3/groups/{group_id}/users{?name,page,per_page,domain_id,description,email,name,enabled,page,per_page}	Lists the users in a specified group.
<b>PUT</b>	/v3/groups/{group_id}/users/{user_id}	Adds a user to a specified group.
<b>DELETE</b>	/v3/groups/{group_id}/users/{user_id}	Removes a user from a group.
<b>HEAD</b>	/v3/groups/{group_id}/users/{user_id}	Validates that a user is in a group.
Credentials		
<b>POST</b>	/v3/credentials	Adds a credential.
<b>GET</b>	/v3/credentials{?page,per_page}	Lists credentials.
<b>GET</b>	/v3/credentials/{credential_id}	Shows details for a specified credential.

Method	URI	Description
<b>PATCH</b>	/v3/credentials/{credential_id}	Updates a specified credential.
<b>DELETE</b>	/v3/credentials/{credential_id}	Deletes a specified credential.
Roles		
<b>POST</b>	/v3/roles	Adds a role.
<b>GET</b>	/v3/roles{?name,page,per_page}	Lists roles.
<b>PUT</b>	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Grants a role to a specified domain user.
<b>HEAD</b>	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Validates that a user has a role on a domain.
<b>DELETE</b>	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Revokes a role from a specified domain user.
<b>GET</b>	/v3/role_assignments{?group.id, role.id, scope.domain.id, scope.project.id, user.id, effective}	Lists role assignments.
Policies		
<b>POST</b>	/v3/policies	Adds a policy.
<b>GET</b>	/v3/policies{?type,page,per_page}	Lists policies.
<b>GET</b>	/v3/policies/{policy_id}	Shows details for a specified policy.
<b>PATCH</b>	/v3/policies/{policy_id}	Updates a specified policy.
<b>DELETE</b>	/v3/policies/{policy_id}	Deletes a specified policy.

## 7.1. API versions

Method	URI	Description
<b>GET</b>	/	Lists information about all Identity API versions.
<b>GET</b>	/v3	Shows details for the Identity API v3.

## 7.1.1. 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)

### 7.1.1.1. Request

This operation does not accept a request body.

### 7.1.1.2. Response

#### Example 7.1. List versions: JSON response

```
{
  "versions": [
    {
      "values": [
        {
          "status": "stable",
          "updated": "2013-03-06T00:00:00Z",
          "media-types": [
            {
              "base": "application/json",
              "type": "application/vnd.openstack.identity-v3+json"
            },
            {
              "base": "application/xml",
              "type": "application/vnd.openstack.identity-v3+xml"
            }
          ],
          "id": "v3.0",
          "links": [
            {
              "href": "http://23.253.228.211:5000/v3/",
              "rel": "self"
            }
          ]
        },
        {
          "status": "stable",
          "updated": "2014-04-17T00:00:00Z",
          "media-types": [
            {
              "base": "application/json",
              "type": "application/vnd.openstack.identity-v2.0+json"
            },
            {
              "base": "application/xml",
              "type": "application/vnd.openstack.identity-v2.0+xml"
            }
          ]
        }
      ]
    }
  ]
}
```

```
        ],
        "id": "v2.0",
        "links": [
            {
                "href": "http://23.253.228.211:5000/v2.0/",
                "rel": "self"
            },
            {
                "href": "http://docs.openstack.org/api/openstack-
identity-service/2.0/content/",
                "type": "text/html",
                "rel": "describedby"
            },
            {
                "href": "http://docs.openstack.org/api/openstack-
identity-service/2.0/identity-dev-guide-2.0.pdf",
                "type": "application/pdf",
                "rel": "describedby"
            }
        ]
    }
}
```

### Example 7.2. List versions: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<versions xmlns="http://docs.openstack.org/identity/api/v2.0">
    <version status="stable" updated="2013-03-06T00:00:00Z" id="v3.0">
        <media-types>
            <media-type base="application/json"
                type="application/vnd.openstack.identity-v3+json"/>
            <media-type base="application/xml"
                type="application/vnd.openstack.identity-v3+xml"/>
        </media-types>
        <links>
            <link href="http://23.253.228.211:5000/v3/" rel="self"/>
        </links>
    </version>
    <version status="stable" updated="2014-04-17T00:00:00Z" id="v2.0">
        <media-types>
            <media-type base="application/json"
                type="application/vnd.openstack.identity-v2.0+json"/>
            <media-type base="application/xml"
                type="application/vnd.openstack.identity-v2.0+xml"/>
        </media-types>
        <links>
            <link href="http://23.253.228.211:5000/v2.0/" rel="self"/>
            <link
                href="http://docs.openstack.org/api/openstack-identity-
service/2.0/content/"
                type="text/html" rel="describedby"/>
            <link
                href="http://docs.openstack.org/api/openstack-identity-
service/2.0/identity-dev-guide-2.0.pdf"
                type="application/pdf" rel="describedby"/>
        </links>
        <link href="http://23.253.228.211:5000/v2.0/" rel="self"/>
        <link
```

```
    href="http://docs.openstack.org/api/openstack-identity-service/2.
0/content/"
        type="text/html" rel="describedby"/>
    <link
        href="http://docs.openstack.org/api/openstack-identity-service/2.
0/identity-dev-guide-2.0.pdf"
        type="application/pdf" rel="describedby"/>
    </version>
</versions>
```

This operation does not return a response body.

## 7.1.2. Show API version details

Method	URI	Description
GET	/v3	Shows details for the Identity API v3.

**Normal response codes:** 200

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

### 7.1.2.1. Request

This operation does not accept a request body.

### 7.1.2.2. Response

#### Example 7.3. Show API version details: JSON response

```
{
  "version": {
    "status": "stable",
    "updated": "2013-03-06T00:00:00Z",
    "media-types": [
      {
        "base": "application/json",
        "type": "application/vnd.openstack.identity-v3+json"
      },
      {
        "base": "application/xml",
        "type": "application/vnd.openstack.identity-v3+xml"
      }
    ],
    "id": "v3.0",
    "links": [
      {
        "href": "http://23.253.228.211:35357/v3/",
        "rel": "self"
      }
    ]
  }
}
```

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

Name	Type	Description
version	String <i>(Required)</i>	A version object.
status	String <i>(Required)</i>	The status of the version. For example, experimental or stable.
updated	String <i>(Required)</i>	The date and time stamp for the last update.

Name	Type	Description
media-types	String <i>(Required)</i>	A media-types object, which shows the supported media types.
id	String <i>(Required)</i>	The version ID, such as v3.0.
links	String <i>(Required)</i>	The version links.

#### Example 7.4. Show API version details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<version xmlns="http://docs.openstack.org/identity/api/v3"
    status="stable" updated="2013-03-06T00:00:00Z" id="v3.0">
    <media-types>
        <media-type base="application/json"
            type="application/vnd.openstack.identity-v3+json"/>
        <media-type base="application/xml"
            type="application/vnd.openstack.identity-v3+xml"/>
    </media-types>
    <links>
        <link href="http://23.253.228.211:35357/v3/" rel="self"/>
    </links>
</version>
```

This operation does not return a response body.

## 7.2. Tokens

Manage tokens.

Method	URI	Description
<b>POST</b>	/v3/auth/tokens	Authenticates and generates a token.
<b>GET</b>	/v3/auth/tokens	Validates a specified token.
<b>HEAD</b>	/v3/auth/tokens	Validates a specified token.
<b>DELETE</b>	/v3/auth/tokens	Revokes a specified token.

## 7.2.1. Authenticate

Method	URI	Description
POST	/v3/auth/tokens	Authenticates and generates a token.

Returns a token, if successful. Each REST request requires the inclusion of a specific authorization token HTTP x-header, defined as X-Auth-Token. Clients obtain X-Auth-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 supplied as 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, an HTTP 400 Bad Request results, 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, a 401 response code is returned.

If the subject token has expired, this call returns a 404 response code.

The Identity API treats expired tokens as not 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 401 Not Authorized error with additional information for the next authentication step.

The following examples illustrate several possible HTTP 401 Unauthorized authentication errors. Other errors like HTTP 403 Forbidden and HTTP 409 Conflict are also possible.

**Normal response codes:** 201

**Error response codes:** identityFault (400, 500, ...), 401, 404, 409, badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

### 7.2.1.1. Request

#### Example 7.5. Authenticate: JSON request

```
{  
    "auth": {  
        "identity": {  
            "methods": [  
                "password"  
            ],  
            "password": {  
                "user": {  
                    "id": "0ca8f6",  
                    "password": "secrete"  
                }  
            }  
        }  
    }  
}
```

#### Example 7.6. Authenticate: JSON request

```
{  
    "auth": {  
        "identity": {  
            "methods": [  
                "password"  
            ],  
            "password": {  
                "user": {  
                    "domain": {  
                        "id": "1789d1"  
                    },  
                    "name": "Joe",  
                    "password": "secrete"  
                }  
            }  
        }  
    }  
}
```

```
        }
    }
}
```

### Example 7.7. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "domain": {
            "name": "example.com"
          },
          "name": "Joe",
          "password": "secrete"
        }
      }
    }
  }
}
```

### Example 7.8. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "token"
      ],
      "token": {
        "id": "e80b74"
      }
    }
  }
}
```

### Example 7.9. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "id": "0ca8f6",
          "password": "secrete"
        }
      }
    },
    "scope": {
      "project": {
        "id": "263fd9"
      }
    }
  }
}
```

```
        }
    }
}
```

### Example 7.10. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "id": "0ca8f6",
          "password": "secrete"
        }
      }
    },
    "scope": {
      "domain": {
        "id": "263fd9"
      }
    }
  }
}
```

### Example 7.11. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "id": "0ca8f6",
          "password": "secrete"
        }
      }
    },
    "scope": {
      "project": {
        "domain": {
          "id": "1789d1"
        },
        "name": "project-x"
      }
    }
  }
}
```

### Example 7.12. Authenticate: JSON request

```
{
  "auth": {
    "identity": {
      "methods": [

```

```

        "password"
    ],
    "password": {
        "user": {
            "id": "0ca8f6",
            "password": "secrete"
        }
    }
},
"scope": {
    "project": {
        "domain": {
            "name": "example.com"
        },
        "name": "project-x"
    }
}
}
}

```

### 7.2.1.2. Response

#### Example 7.13. Authenticate: JSON response

```
{
    "token": {
        "expires_at": "2013-02-27T18:30:59.999999Z",
        "issued_at": "2013-02-27T16:30:59.999999Z",
        "methods": [
            "password"
        ],
        "user": {
            "domain": {
                "id": "1789d1",
                "links": {
                    "self": "http://identity:35357/v3/domains/1789d1"
                },
                "name": "example.com"
            },
            "id": "0ca8f6",
            "links": {
                "self": "http://identity:35357/v3/users/0ca8f6"
            },
            "name": "Joe"
        }
    }
}
```

#### Example 7.14. Authenticate: JSON response

```
{
    "token": {
        "catalog": "FIXME(dolph): need an example here",
        "expires_at": "2013-02-27T18:30:59.999999Z",
        "issued_at": "2013-02-27T16:30:59.999999Z",
        "methods": [
            "password"
        ],
        "project": {

```

```
        "domain": {
            "id": "1789d1",
            "links": {
                "self": "http://identity:35357/v3/domains/1789d1"
            },
            "name": "example.com"
        },
        "id": "263fd9",
        "links": {
            "self": "http://identity:35357/v3/projects/263fd9"
        },
        "name": "project-x"
    },
    "roles": [
        {
            "id": "76e72a",
            "links": {
                "self": "http://identity:35357/v3/roles/76e72a"
            },
            "name": "admin"
        },
        {
            "id": "f4f392",
            "links": {
                "self": "http://identity:35357/v3/roles/f4f392"
            },
            "name": "member"
        }
    ],
    "user": {
        "domain": {
            "id": "1789d1",
            "links": {
                "self": "http://identity:35357/v3/domains/1789d1"
            },
            "name": "example.com"
        },
        "id": "0ca8f6",
        "links": {
            "self": "http://identity:35357/v3/users/0ca8f6"
        },
        "name": "Joe"
    }
}
}
```

### Example 7.15. Authenticate: JSON response

```
{
    "token": {
        "catalog": "FIXME(dolph): need an example here",
        "expires_at": "2013-02-27T18:30:59.999999Z",
        "issued_at": "2013-02-27T16:30:59.999999Z",
        "methods": [
            "password"
        ],
        "domain": {
            "id": "1789d1",
            "links": {
                "self": "http://identity:35357/v3/domains/1789d1"
            }
        }
    }
}
```

```
        },
        "name": "example.com"
    },
    "roles": [
        {
            "id": "76e72a",
            "links": {
                "self": "http://identity:35357/v3/roles/76e72a"
            },
            "name": "admin"
        },
        {
            "id": "f4f392",
            "links": {
                "self": "http://identity:35357/v3/roles/f4f392"
            },
            "name": "member"
        }
    ],
    "user": {
        "domain": {
            "id": "1789d1",
            "links": {
                "self": "http://identity:35357/v3/domains/1789d1"
            },
            "name": "example.com"
        },
        "id": "0ca8f6",
        "links": {
            "self": "http://identity:35357/v3/users/0ca8f6"
        },
        "name": "Joe"
    }
}
```

## 7.2.2. Validate token

Method	URI	Description
GET	/v3/auth/tokens	Validates a specified token.

Pass your own token in the X-Auth-Token header and 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, 203

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

### 7.2.2.1. 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 7.16. Validate token: JSON request

```
Headers:
X-Auth-Token: 1dd7e3
X-Subject-Token: c67580
```

This operation does not accept a request body.

### 7.2.2.2. Response

#### Example 7.17. Validate token: JSON response

```
{
  "token": {
    "expires_at": "2013-02-27T18:30:59.999999Z",
    "issued_at": "2013-02-27T16:30:59.999999Z",
    "methods": [
      "password"
    ],
    "user": {
      "domain": {
        "id": "1789d1",
        "links": {
          "self": "http://identity:35357/v3/domains/1789d1"
        },
        "name": "example.com"
      }
    }
  }
}
```

```
        "id": "0ca8f6",
        "links": {
            "self": "http://identity:35357/v3/users/0ca8f6"
        },
        "name": "Joe"
    }
}
```

## 7.2.3. Check token

Method	URI	Description
HEAD	/v3/auth/tokens	Validates a specified 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. A 204 response indicates that the X-Subject-Token is valid.

**Normal response codes:** 204

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

### 7.2.3.1. 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 7.18. Check token: JSON request

```
Headers:
X-Auth-Token: 1dd7e3
X-Subject-Token: c67580
```

This operation does not accept a request body.

## 7.2.4. Revoke token

Method	URI	Description
<b>DELETE</b>	/v3/auth/tokens	Revokes a specified 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). An additional `X-Auth-Token` is not required.

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

### 7.2.4.1. 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 7.19. Revoke token: JSON request

```
Headers:
X-Auth-Token: 1dd7e3
X-Subject-Token: c67580
```

This operation does not accept a request body.

## 7.3. Service catalog

Manage the catalog of services.

Method	URI	Description
<b>POST</b>	/v3/services	Adds a service.
<b>GET</b>	/v3/services{?type,page,per_page}	Lists services.
<b>GET</b>	/v3/services/{service_id}	Shows details for a specified service.
<b>PATCH</b>	/v3/services/{service_id}	Updates a specified service.
<b>DELETE</b>	/v3/services/{service_id}	Deletes a specified service.

## 7.3.1. Add service

Method	URI	Description
POST	/v3/services	Adds a service.

**Normal response codes:** 201

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

### 7.3.1.1. Request

This table shows the header parameters for the add service request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 7.20. Add service: JSON request

```
{
    "type": "volume"
}
```

### 7.3.1.2. Response

#### Example 7.21. Add service: JSON response

```
{
    "service": {
        "id": "--service-id--",
        "type": "volume"
    }
}
```

## 7.3.2. List services

Method	URI	Description
GET	/v3/services{?type,page,per_page}	Lists services.

**Normal response codes:** 200

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

### 7.3.2.1. 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.
page	String <i>(Optional)</i>	Enables you to page through the list.
per_page	String <i>(Optional)</i>	Sets the page size for paging through the list. Default page size is 30.

This operation does not accept a request body.

### 7.3.2.2. Response

#### Example 7.22. List services: JSON response

```
{
  "links": {
    "next": null,
    "previous": null,
    "self": "http://identity:5000/v3/services"
  },
  "services": [
    {
      "description": "Keystone Identity Service",
      "id": "--service-id--",
      "links": {
        "self": "http://identity:5000/v3/services/--service-id--"
      },
      "name": "keystone",
      "type": "identity"
    }
  ]
}
```

```
{  
    "id": "--service-id--",  
    "links": {  
        "self": "http://identity:5000/v3/services/--service-id--"  
    },  
    "type": "volume"  
}  
]  
}
```

### 7.3.3. Show service details

Method	URI	Description
GET	/v3/services/{service_id}	Shows details for a specified service.

**Normal response codes:** 200

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

#### 7.3.3.1. 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.

#### 7.3.3.2. Response

##### Example 7.23. Show service details: JSON response

```
{
  "service": {
    "description": "Keystone Identity Service",
    "id": "--service-id--",
    "links": {
      "self": "http://identity:5000/v3/services/--service-id--"
    },
    "name": "keystone",
    "type": "identity"
  }
}
```

## 7.3.4. Update service

Method	URI	Description
PATCH	/v3/services/{service_id}	Updates a specified service.

**Normal response codes:** 200

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

### 7.3.4.1. 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 7.24. Update service: JSON request

```
{
    "type": "volume"
}
```

### 7.3.4.2. Response

#### Example 7.25. Update service: JSON response

```
{
    "service": {
        "id": "--service-id--",
        "type": "volume"
    }
}
```

## 7.3.5. Delete service

Method	URI	Description
<b>DELETE</b>	/v3/services/{service_id}	Deletes a specified service.



### Warning

Deleting a service when endpoints exist should either (1) delete all associated endpoints or (2) fail until endpoints are deleted.

**Normal response codes:** 204

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

### 7.3.5.1. 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.

## 7.4. Endpoints

Manage endpoints.

Method	URI	Description
<b>POST</b>	/v3/endpoints	Adds an endpoint.
<b>GET</b>	/v3/endpoints{?interface, service_id,page,per_page}	Lists available endpoints.
<b>PATCH</b>	/v3/endpoints/{endpoint_id}	Updates a specified endpoint.
<b>DELETE</b>	/v3/endpoints/{endpoint_id}	Deletes a specified endpoint.

## 7.4.1. Add endpoint

Method	URI	Description
POST	/v3/endpoints	Adds an endpoint.

**Normal response codes:** 201

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

### 7.4.1.1. Request

This table shows the header parameters for the add endpoint request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 7.26. Add endpoint: JSON request

```
{
  "endpoint": {
    "interface": "[admin|public|internal]",
    "name": "name",
    "region": "--optional--",
    "url": "...",
    "service_id": "--service-id--"
  }
}
```

### 7.4.1.2. Response

#### Example 7.27. Add endpoint: JSON response

```
{
  "endpoint": {
    "id": "--endpoint-id--",
    "interface": "internal",
    "links": {
      "self": "http://identity:35357/v3/endpoints/--endpoint-id--"
    },
    "name": "the internal volume endpoint",
    "region": "--optional--",
    "service_id": "--service-id--",
    "url": ...
  }
}
```

## 7.4.2. List endpoints

Method	URI	Description
GET	/v3/endpoints{?interface,service_id,page,per_page}	Lists available endpoints.

**Normal response codes:** 200

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

### 7.4.2.1. 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.
page	String <i>(Optional)</i>	Enables you to page through the list.
per_page	String <i>(Optional)</i>	Sets the page size for paging through the list. Default page size is 30.

This operation does not accept a request body.

### 7.4.2.2. Response

#### Example 7.28. List endpoints: JSON response

```
[
  {
    "id": "--endpoint-id--",
    "interface": "public",
    "links": {
      "self": "http://identity:35357/v3/endpoints/--endpoint-id--"
    },
    "name": "the public volume endpoint",
    "service_id": "--service-id--"
  },
  {
    "id": "--endpoint-id--",
    "links": {
      "self": "http://identity:35357/v3/endpoints/--endpoint-id--"
    }
  }
]
```

```
        "interface": "internal",
        "links": [
            "self": "http://identity:35357/v3/endpoints/--endpoint-id--"
        ],
        "name": "the internal volume endpoint",
        "service_id": "--service-id--"
    }
]
```

## 7.4.3. Update endpoint

Method	URI	Description
PATCH	/v3/endpoints/{endpoint_id}	Updates a specified endpoint.

**Normal response codes:** 200

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

### 7.4.3.1. 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 7.29. Update endpoint: JSON request

```
{
  "endpoint": {
    "interface": "--optional--",
    "name": "--optional--",
    "region": "--optional--",
    "url": "--optional--",
    "service_id": "--optional--"
  }
}
```

### 7.4.3.2. Response

#### Example 7.30. Update endpoint: JSON response

```
{
  "endpoint": {
    "id": "--endpoint-id--",
    "interface": "internal",
    "links": {
      "self": "http://identity:35357/v3/endpoints/--endpoint-id--"
    },
    "name": "the internal volume endpoint",
    "region": "--optional--",
    "service_id": "--service-id--",
    "url": "..."
  }
}
```



## 7.4.4. Delete endpoint

Method	URI	Description
DELETE	/v3/endpoints/{endpoint_id}	Deletes a specified endpoint.

**Normal response codes:** 204

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

### 7.4.4.1. 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.

## 7.5. Domains

Manage 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	Adds a domain.

Method	URI	Description
<b>GET</b>	/v3/domains{?name,enabled,page,per_page}	Lists domains.
<b>GET</b>	/v3/domains/{domain_id}	Shows details for a specified domain.
<b>PATCH</b>	/v3/domains/{domain_id}	Updates a specified domain.
<b>DELETE</b>	/v3/domains/{domain_id}	Deletes a specified domain.
<b>GET</b>	/v3/domains/{domain_id}/users/{user_id}/roles	Lists roles for a user on a domain.
<b>GET</b>	/v3/domains/{domain_id}/groups/{group_id}/roles	Lists roles for a specified domain group.
<b>PUT</b>	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Grants a specified role to a specified domain group.
<b>HEAD</b>	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Validates that a group has a role on a domain.
<b>DELETE</b>	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Revokes a role from a group on a domain.

## 7.5.1. Add domain

Method	URI	Description
POST	/v3/domains	Adds a domain.

**Normal response codes:** 201

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

### 7.5.1.1. Request

This table shows the header parameters for the add domain request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 7.31. Add domain: JSON request

```
{
  "domain": {
    "description": "--optional--",
    "enabled": "--optional--",
    "name": "..."
  }
}
```

### 7.5.1.2. Response

#### Example 7.32. Add domain: JSON response

```
{
  "domain": {
    "description": "desc of domain",
    "enabled": true,
    "id": "--domain-id--",
    "links": {
      "self": "http://identity:35357/v3/domains/--domain-id--"
    },
    "name": "my domain"
  }
}
```

## 7.5.2. List domains

Method	URI	Description
GET	/v3/domains{?name,enabled,page,per_page}	Lists domains.

**Normal response codes:** 200

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

### 7.5.2.1. Request

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.
page	String <i>(Optional)</i>	Enables you to page through the list.
per_page	String <i>(Optional)</i>	Sets the page size for paging through the list. Default page size is 30.

This operation does not accept a request body.

### 7.5.2.2. Response

#### Example 7.33. List domains: JSON response

```
{
  "domains": [
    {
      "description": "desc of domain",
      "enabled": true,
      "id": "--domain-id--",
      "links": {
        "self": "http://identity:35357/v3/domains/--domain-id--"
      },
      "name": "my domain"
    },
    {
      "description": "desc of another domain",
      "enabled": true,
      "id": "--domain-id--",
      "links": {
        "self": "http://identity:35357/v3/domains/--domain-id--"
      },
      "name": "another domain"
    }
  ]
}
```

```
        }  
    ]  
}
```

## 7.5.3. Show domain details

Method	URI	Description
GET	/v3/domains/{domain_id}	Shows details for a specified domain.

**Normal response codes:** 200

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

### 7.5.3.1. Request

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

Name	Type	Description
{domain_id}	Uuid	The domain ID.

This operation does not accept a request body.

### 7.5.3.2. Response

#### Example 7.34. Show domain details: JSON response

```
{
  "domain": {
    "description": "desc of domain",
    "enabled": true,
    "id": "--domain-id--",
    "links": {
      "self": "http://identity:35357/v3/domains/--domain-id--"
    },
    "name": "my domain"
  }
}
```

## 7.5.4. Update domain

Method	URI	Description
PATCH	/v3/domains/{domain_id}	Updates a specified domain.

**Normal response codes:** 200

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

### 7.5.4.1. 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 7.35. Update domain: JSON request

```
{
  "domain": {
    "description": "my updated domain",
    "enabled": true,
    "name": "myUpdatedDomain"
  }
}
```

### 7.5.4.2. Response

#### Example 7.36. Update domain: JSON response

```
{
  "domain": {
    "description": "my updated domain",
    "enabled": true,
    "id": "--domain-id--",
    "links": {
      "self": "http://identity:35357/v3/domains/--domain-id--"
    },
    "name": "myUpdatedDomain"
  }
}
```

## 7.5.5. Delete domain

Method	URI	Description
DELETE	/v3/domains/{domain_id}	Deletes a specified domain.

To minimize the risk of accidentally deleting a domain, you must first disable the domain by using the update domain API. If you try to delete an enabled domain, the call returns an HTTP 403 Forbidden response.



### Warning

Deleting a domain deletes all entities owned by it, such as users, groups, and projects, as well as any credentials and granted roles that relate to those entities.

**Normal response codes:** 204

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

### 7.5.5.1. Request

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

Name	Type	Description
{domain_id}	Uuid	The domain ID.

This operation does not accept a request body.

## 7.5.6. 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:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

### 7.5.6.1. Request

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.

### 7.5.6.2. Response

#### Example 7.37. List roles for domain user: JSON response

```
[  
  {  
    "id": "--role-id--",  
    "name": "--role-name--"  
  },  
  {  
    "id": "--role-id--",  
    "name": "--role-name--"  
  }  
]
```

## 7.5.7. List roles for domain group

Method	URI	Description
GET	/v3/domains/{domain_id}/groups/{group_id}/roles	Lists roles for a specified domain group.

**Normal response codes:** 200

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

### 7.5.7.1. Request

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.

### 7.5.7.2. Response

#### Example 7.38. List roles for domain group: JSON response

```
[  
  {  
    "id": "--role-id--",  
    "name": "--role-name--"  
  },  
  {  
    "id": "--role-id--",  
    "name": "--role-name--"  
  }  
]
```

## 7.5.8. Grant role to domain group

Method	URI	Description
PUT	/v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}	Grants a specified role to a specified domain group.

**Normal response codes:** 204

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

### 7.5.8.1. Request

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.

## 7.5.9. 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:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

### 7.5.9.1. Request

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.

## 7.5.10. Revoke role from domain group

Method	URI	Description
<b>DELETE</b>	/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:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

### 7.5.10.1. Request

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.

## 7.6. Projects

Manage projects.

Method	URI	Description
<b>POST</b>	/v3/projects	Adds a project.
<b>GET</b>	/v3/projects{?domain_id,name,enabled,page,per_page}	Lists projects.
<b>GET</b>	/v3/projects/{project_id}	Shows details for a specified project.
<b>PATCH</b>	/v3/projects/{project_id}	Updates a specified project.
<b>DELETE</b>	/v3/projects/{project_id}	Deletes a specified project.
<b>GET</b>	/v3/projects/{project_id}/users/{user_id}/roles	Lists roles for a user in a project.
<b>PUT</b>	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Grants a role to a user on a project.
<b>HEAD</b>	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Validates that a user has a specified role on a project.
<b>DELETE</b>	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Revokes a role from a project user.
<b>GET</b>	/v3/projects/{project_id}/groups/{group_id}/roles	Lists roles for a project group.
<b>PUT</b>	/v3/projects/{project_id}/groups/{group_id}/roles	Grants a role to a project group.
<b>HEAD</b>	/v3/projects/{project_id}/groups/{group_id}/roles	Validates that a project group has a role.
<b>DELETE</b>	/v3/projects/{project_id}/groups/{group_id}/roles	Revokes a role from a project group.



## 7.6.1. Add project

Method	URI	Description
POST	/v3/projects	Adds a project.

**Normal response codes:** 201

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

### 7.6.1.1. Request

This table shows the header parameters for the add project request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 7.39. Add project: JSON request

```
{
  "project": {
    "description": "...",
    "domain_id": "...",
    "enabled": "...",
    "name": "..."
  }
}
```

### 7.6.1.2. Response

#### Example 7.40. Add project: JSON response

```
{
  "project": {
    "domain_id": "--domain-id--",
    "enabled": true,
    "id": "--project-id--",
    "links": {
      "self": "http://identity:35357/v3/projects/--project-id--"
    },
    "name": "a project name"
  }
}
```

## 7.6.2. List projects

Method	URI	Description
GET	/v3/projects{?domain_id,name(enabled,page,per_page)}	Lists projects.

**Normal response codes:** 200

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

### 7.6.2.1. 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.
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.
page	String <i>(Optional)</i>	Enables you to page through the list.
per_page	String <i>(Optional)</i>	Sets the page size for paging through the list. Default page size is 30.

This operation does not accept a request body.

### 7.6.2.2. Response

#### Example 7.41. List projects: JSON response

```
{
  "projects": [
    {
      "domain_id": "--domain-id--",
      "enabled": true,
      "id": "--project-id--",
      "links": {
        "self": "http://identity:35357/v3/projects/--project-id--"
      },
    }
  ]
}
```

```
        "name": "a project name"
    },
    {
        "domain_id": "--domain-id--",
        "enabled": true,
        "id": "--project-id--",
        "links": [
            {
                "self": "http://identity:35357/v3/projects/--project-id--"
            },
            {
                "name": "another project"
            }
        ],
        "links": [
            {
                "self": "http://identity:35357/v3/projects",
                "previous": null,
                "next": null
            }
        ]
    }
```

## 7.6.3. Show project details

Method	URI	Description
GET	/v3/projects/{project_id}	Shows details for a specified project.

**Normal response codes:** 200

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

### 7.6.3.1. 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.

### 7.6.3.2. Response

#### Example 7.42. Show project details: JSON response

```
{
  "project": {
    "domain_id": "--domain-id--",
    "enabled": true,
    "id": "--project-id--",
    "links": {
      "self": "http://identity:35357/v3/projects/--project-id--"
    },
    "name": "a project name"
  }
}
```

## 7.6.4. Update project

Method	URI	Description
PATCH	/v3/projects/{project_id}	Updates a specified project.

**Normal response codes:** 200

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

### 7.6.4.1. 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 7.43. Update project: JSON request

```
{
  "project": {
    "description": "my updated project",
    "domain_id": "...",
    "enabled": true,
    "name": "myUpdatedProject"
  }
}
```

### 7.6.4.2. Response

#### Example 7.44. Update project: JSON response

```
{
  "project": {
    "description": "my updated project",
    "domain_id": "--domain-id--",
    "enabled": true,
    "id": "--project-id--",
    "links": {
      "self": "http://identity:35357/v3/projects/--project-id--"
    },
    "name": "myUpdatedProject"
  }
}
```

## 7.6.5. Delete project

Method	URI	Description
DELETE	/v3/projects/{project_id}	Deletes a specified project.

**Normal response codes:** 204

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

### 7.6.5.1. 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.

## 7.6.6. 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:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

### 7.6.6.1. 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.

### 7.6.6.2. Response

#### Example 7.45. List roles for project user: JSON response

```
[  
  {  
    "id": "--role-id--",  
    "name": "--role-name--"  
  },  
  {  
    "id": "--role-id--",  
    "name": "--role-name--"  
  }  
]
```

## 7.6.7. 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 on a project.

**Normal response codes:** 204

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

### 7.6.7.1. 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.

## 7.6.8. 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 specified role on a project.

**Normal response codes:** 204

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

### 7.6.8.1. 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.

## 7.6.9. Revoke role from project user

Method	URI	Description
<b>DELETE</b>	/v3/projects/{project_id}/users/{user_id}/roles/{role_id}	Revokes a role from a project 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)

### 7.6.9.1. 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.

## 7.6.10. 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:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

### 7.6.10.1. 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.

### 7.6.10.2. Response

#### Example 7.46. List roles for project group: JSON response

```
[  
  {  
    "id": "--role-id--",  
    "name": "--role-name--"  
  },  
  {  
    "id": "--role-id--",  
    "name": "--role-name--"  
  }  
]
```

## 7.6.11. Grant role to project group

Method	URI	Description
PUT	/v3/projects/{project_id}/groups/{group_id}/roles	Grants a role to a project group.

**Normal response codes:** 204

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

### 7.6.11.1. 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.

This operation does not accept a request body.

## 7.6.12. Check role for project group

Method	URI	Description
HEAD	/v3/projects/{project_id}/groups/{group_id}/roles	Validates that a project group has 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)

### 7.6.12.1. 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.

This operation does not accept a request body.

## 7.6.13. Revoke Role from Project Group

Method	URI	Description
<b>DELETE</b>	/v3/projects/{project_id}/groups/{group_id}/roles	Revokes a role from a project group.

**Normal response codes:** 204

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

### 7.6.13.1. 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.

This operation does not accept a request body.

## 7.7. Users

Manage users.

Method	URI	Description
<b>POST</b>	/v3/users	Adds a user.
<b>GET</b>	/v3/users{?domain_id,email,name,enabled,page,per_page}	Lists users.
<b>GET</b>	/v3/users/{user_id}	Shows details for a specified user.
<b>PATCH</b>	/v3/users/{user_id}	Updates the password for or enables or disables a specified user.
<b>DELETE</b>	/v3/users/{user_id}	Deletes a specified user.
<b>GET</b>	/v3/users/{user_id}/groups	Lists groups for a specified user.
<b>GET</b>	/v3/users/{user_id}/projects	List projects for a specified user.
<b>GET</b>	/v3/users/{user_id}/roles{?name,project_id, domain_id}	Lists roles for a specified user.

## 7.7.1. Add user

Method	URI	Description
POST	/v3/users	Adds 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)

### 7.7.1.1. Request

This table shows the header parameters for the add user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 7.47. Add user: JSON request

```
{
  "user": {
    "default_project_id": "...",
    "description": "...",
    "domain_id": "--optional--",
    "email": "...",
    "enabled": "...",
    "name": "...",
    "password": "--optional--"
  }
}
```

### 7.7.1.2. Response

#### Example 7.48. Add user: JSON response

```
{
  "user": {
    "default_project_id": "--default-project-id--",
    "description": "a user",
    "domain_id": "1789d1",
    "email": "...",
    "enabled": true,
    "id": "--user-id--",
    "links": {
      "self": "http://identity:35357/v3/users/--user-id--"
    },
    "name": "admin"
  }
}
```

## 7.7.2. List users

Method	URI	Description
GET	/v3/users{?domain_id,email,name,enabled,page,per_page}	Lists users.

**Normal response codes:** 200

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

### 7.7.2.1. 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.
email	String <i>(Optional)</i>	Filters on email.
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.
page	String <i>(Optional)</i>	Enables you to page through the list.
per_page	String <i>(Optional)</i>	Sets the page size for paging through the list. Default page size is 30.

This operation does not accept a request body.

### 7.7.2.2. Response

#### Example 7.49. List users: JSON response

```
{
  "users": [
    {
      "default_project_id": "--default-project-id--",
      "description": "a user",
      "domain_id": "1789d1",
      "email": "user@example.com",
      "enabled": true,
      "name": "User"
    }
  ]
}
```

```
        "email": "...",
        "enabled": true,
        "id": "--user-id--",
        "links": {
            "self": "http://identity:35357/v3/users/--user-id--"
        },
        "name": "admin"
    },
    {
        "default_project_id": "--default-project-id--",
        "description": "another user",
        "domain_id": "1789d1",
        "email": "...",
        "enabled": true,
        "id": "--user-id--",
        "links": {
            "self": "http://identity:35357/v3/users/--user-id--"
        },
        "name": "someone"
    }
],
"links": {
    "self": "http://identity:35357/v3/users",
    "previous": null,
    "next": null
}
}
```

## 7.7.3. Show user details

Method	URI	Description
GET	/v3/users/{user_id}	Shows details for a specified user.

**Normal response codes:** 200

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

### 7.7.3.1. 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.

### 7.7.3.2. Response

#### Example 7.50. Show user details: JSON response

```
{
  "user": {
    "default_project_id": "--default-project-id--",
    "description": "a user",
    "domain_id": "1789d1",
    "email": "...",
    "enabled": true,
    "id": "--user-id--",
    "links": {
      "self": "http://identity:35357/v3/users/--user-id--"
    },
    "name": "admin"
  }
}
```

## 7.7.4. Update user

Method	URI	Description
PATCH	/v3/users/{user_id}	Updates the password for or enables or disables a specified user.

This operation might return the HTTP 501 Not Implemented code if the back-end driver does not allow this functionality.

**Normal response codes:** 200

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

### 7.7.4.1. 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 7.51. Update user: JSON request

```
{
  "user": {
    "default_project_id": "...",
    "description": "my updated user",
    "email": "...",
    "enabled": true
  }
}
```

### 7.7.4.2. Response

#### Example 7.52. Update user: JSON response

```
{
  "user": {
    "default_project_id": "--default-project-id--",
    "description": "my updated user",
    "domain_id": "1789d1",
    "email": "...",
    "enabled": true,
    "id": "--user-id--",
    "links": {
      "self": "http://identity:35357/v3/users/--user-id--"
    }
  }
}
```

```
        },
        "name": "admin"
    }
}
```

## 7.7.5. Delete user

Method	URI	Description
DELETE	/v3/users/{user_id}	Deletes a specified 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)

### 7.7.5.1. 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.

## 7.7.6. List groups for user

Method	URI	Description
GET	/v3/users/{user_id}/groups	Lists groups for a specified user.

**Normal response codes:** 200

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

### 7.7.6.1. 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.

### 7.7.6.2. Response

#### Example 7.53. List groups for user: JSON response

```
{
  "groups": [
    {
      "description": "Developers cleared for work on all general projects",
      "domain_id": "--domain-id--",
      "id": "--group-id--",
      "links": {
        "self": "http://identity:35357/v3/groups/--group-id--"
      },
      "name": "Developers"
    },
    {
      "description": "Developers cleared for work on secret projects",
      "domain_id": "--domain-id--",
      "id": "--group-id--",
      "links": {
        "self": "http://identity:35357/v3/groups/--group-id--"
      },
      "name": "Secure Developers"
    }
  ],
  "links": {
    ...
  }
}
```

```
        "self": "http://identity:35357/v3/users/--user-id--/groups",
        "previous": null,
        "next": null
    }
}
```

## 7.7.7. List projects for user

Method	URI	Description
GET	/v3/users/{user_id}/projects	List projects for a specified user.

**Normal response codes:** 200

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

### 7.7.7.1. 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.

### 7.7.7.2. Response

#### Example 7.54. List projects for user: JSON response

```
{
  "projects": [
    {
      "domain_id": "--domain-id--",
      "enabled": true,
      "id": "--project-id--",
      "links": {
        "self": "http://identity:35357/v3/projects/--project-id--"
      },
      "name": "a project name"
    },
    {
      "domain_id": "--domain-id--",
      "enabled": true,
      "id": "--project-id--",
      "links": {
        "self": "http://identity:35357/v3/projects/--project-id--"
      },
      "name": "another domain"
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/users/--user-id--/projects",
    "next": "http://identity:35357/v3/users/12345678901234567890123456789012/projects?marker=--project-id--"
  }
}
```

```
        "previous": null,  
        "next": null  
    }  
}
```

## 7.7.8. List roles for user

Method	URI	Description
GET	/v3/users/{user_id}/roles{?name, project_id, domain_id}	Lists roles for a specified user.

**Normal response codes:** 200

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

### 7.7.8.1. 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
{user_id}	Uuid	The user ID.

This operation does not accept a request body.

### 7.7.8.2. Response

#### Example 7.55. List roles for user: JSON response

```
[
  {
    "id": "--role-id--",
    "name": "--role-name--",
    "project_id": "--project-id--"
  },
  {
    "domain_id": "--domain-id--",
    "id": "--role-id--",
    "name": "--role-name--"
  }
]
```

## 7.8. Groups

Manage groups.

Method	URI	Description
POST	/v3/groups	Adds a group.

Method	URI	Description
<b>GET</b>	/v3/groups{?domain_id,page,per_page}	Lists groups.
<b>GET</b>	/v3/groups/{group_id}	Shows details for a specified group.
<b>PATCH</b>	/v3/groups/{group_id}	Updates a specified group.
<b>DELETE</b>	/v3/groups/{group_id}	Deletes a specified group.
<b>GET</b>	/v3/groups/{group_id}/users{?name,page,per_page,domain_id,description,email,name(enabled),page,per_page}	Lists the users in a specified group.
<b>PUT</b>	/v3/groups/{group_id}/users/{user_id}	Adds a user to a specified group.
<b>DELETE</b>	/v3/groups/{group_id}/users/{user_id}	Removes a user from a group.
<b>HEAD</b>	/v3/groups/{group_id}/users/{user_id}	Validates that a user is in a group.

## 7.8.1. Add group

Method	URI	Description
POST	/v3/groups	Adds a group.

**Normal response codes:** 201

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

### 7.8.1.1. Request

#### Example 7.56. Add group: JSON request

```
{  
    "group": {  
        "description": "--optional--",  
        "domain_id": "--optional--",  
        "name": "..."  
    }  
}
```

### 7.8.1.2. Response

#### Example 7.57. Add group: JSON response

```
{  
    "group": {  
        "description": "Developers cleared for work on secret projects",  
        "domain_id": "--domain-id--",  
        "id": "--group-id--",  
        "links": {  
            "self": "http://identity:35357/v3/groups/--group-id--"  
        },  
        "name": "Secure Developers"  
    }  
}
```

## 7.8.2. List groups

Method	URI	Description
GET	/v3/groups{?domain_id,page,per_page}	Lists groups.

**Normal response codes:** 200

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

### 7.8.2.1. 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.
page	String <i>(Optional)</i>	Enables you to page through the list.
per_page	String <i>(Optional)</i>	Sets the page size for paging through the list. Default page size is 30.

This operation does not accept a request body.

### 7.8.2.2. Response

#### Example 7.58. List groups: JSON response

```
{
  "groups": [
    {
      "description": "Developers cleared for work on all general projects",
      "domain_id": "--domain-id--",
      "id": "--group-id--",
      "links": {
        "self": "http://identity:35357/v3/groups/--group-id--"
      },
      "name": "Developers"
    },
    {
      "description": "Developers cleared for work on secret projects",
      "domain_id": "--domain-id--",
      "id": "--group-id--",
      "links": {
        "self": "http://identity:35357/v3/groups/--group-id--"
      },
      "name": "Secret Developers"
    }
  ]
}
```

```
        "id": "--group-id--",
        "links": {
            "self": "http://identity:35357/v3/groups/--group-id--"
        },
        "name": "Secure Developers"
    },
    {
        "description": "Testers cleared for work on all general projects",
        "domain_id": "--domain-id--",
        "id": "--group-id--",
        "links": {
            "self": "http://identity:35357/v3/groups/--group-id--"
        },
        "name": "Testers"
    }
],
"links": {
    "self": "http://identity:35357/v3/groups",
    "previous": null,
    "next": null
}
}
```

## 7.8.3. Show group details

Method	URI	Description
GET	/v3/groups/{group_id}	Shows details for a specified group.

**Normal response codes:** 200

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

### 7.8.3.1. Request

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

Name	Type	Description
{group_id}	Uuid	The group ID.

This operation does not accept a request body.

### 7.8.3.2. Response

#### Example 7.59. Show group details: JSON response

```
{
  "group": {
    "description": "Developers cleared for work on secret projects",
    "domain_id": "--domain-id--",
    "id": "--group-id--",
    "links": {
      "self": "http://identity:35357/v3/groups/--group-id--"
    },
    "name": "Secure Developers"
  }
}
```

## 7.8.4. Update group

Method	URI	Description
PATCH	/v3/groups/{group_id}	Updates a specified group.

Use this operation to update the name or description of a group. This operation might return the HTTP 501 Not Implemented code if the back-end driver does not allow this functionality.

**Normal response codes:** 200

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

### 7.8.4.1. Request

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

Name	Type	Description
{group_id}	Uuid	The group ID.

#### Example 7.60. Update group: JSON request

```
{
  "group": {
    "description": "my updated group",
    "name": "myUpdatedGroup"
  }
}
```

### 7.8.4.2. Response

#### Example 7.61. Update group: JSON response

```
{
  "group": {
    "description": "my updated group",
    "domain_id": "--domain-id--",
    "id": "--group-id--",
    "links": {
      "self": "http://identity:35357/v3/groups/--group-id--"
    },
    "name": "myUpdatedGroup"
  }
}
```

## 7.8.5. Delete group

Method	URI	Description
DELETE	/v3/groups/{group_id}	Deletes a specified group.

**Normal response codes:** 204

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

### 7.8.5.1. Request

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

Name	Type	Description
{group_id}	Uuid	The group ID.

This operation does not accept a request body.

## 7.8.6. List users in group

Method	URI	Description
<b>GET</b>	/v3/groups/{group_id}/users{?name,page,per_page,domain_id,description,email,name(enabled,page,per_page)}	Lists the users in a specified group.

**Normal response codes:** 200

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

### 7.8.6.1. Request

This table shows the URI parameters for the list users in group request:

Name	Type	Description
{group_id}	Uuid	The group ID.

This table shows the query parameters for the list users in group request:

Name	Type	Description
name	String (Optional)	Filters on a name.
page	String (Optional)	Enables you to page through the list.
per_page	String (Optional)	Sets the page size for paging through the list. Default page size is 30.
domain_id	String (Optional)	Filters on domain_id.
description	String (Optional)	Filters on a description.
email	String (Optional)	Filters on email.
name	String (Optional)	Filters on a name.
enabled	String (Optional)	Filters on enabled or disabled roles. Values are true or false.
page	String (Optional)	Enables you to page through the list.
per_page	String (Optional)	Sets the page size for paging through the list. Default page size is 30.

This operation does not accept a request body.

## 7.8.6.2. Response

### Example 7.62. List users in group: JSON response

```
{  
    "users": [  
        {  
            "default_project_id": "--default-project-id--",  
            "description": "a user",  
            "domain_id": "--domain-id--",  
            "email": "...",  
            "enabled": true,  
            "id": "--user-id--",  
            "links": {  
                "self": "http://identity:35357/v3/users/--user-id--"  
            },  
            "name": "admin"  
        },  
        {  
            "default_project_id": "--default-project-id--",  
            "description": "another user",  
            "domain_id": "--domain-id--",  
            "email": "...",  
            "enabled": true,  
            "id": "--user-id--",  
            "links": {  
                "self": "http://identity:35357/v3/users/--user-id--"  
            },  
            "name": "someone"  
        }  
    ],  
    "links": {  
        "self": "http://identity:35357/v3/groups/--group-id--/users",  
        "previous": null,  
        "next": null  
    }  
}
```

## 7.8.7. Add user to group

Method	URI	Description
PUT	/v3/groups/{group_id}/users/{user_id}	Adds a user to a specified group.

**Normal response codes:** 204

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

### 7.8.7.1. Request

This table shows the URI parameters for the add user to group request:

Name	Type	Description
{group_id}	Uuid	The group ID.
{user_id}	Uuid	The user ID.

This operation does not accept a request body.

## 7.8.8. Remove user from group

Method	URI	Description
<b>DELETE</b>	/v3/groups/{group_id}/users/{user_id}	Removes a user from a group.

**Normal response codes:** 204

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

### 7.8.8.1. Request

This table shows the URI parameters for the remove user from group request:

Name	Type	Description
{group_id}	Uuid	The group ID.
{user_id}	Uuid	The user ID.

This operation does not accept a request body.

## 7.8.9. Check user membership in group

Method	URI	Description
<b>HEAD</b>	/v3/groups/{group_id}/users/{user_id}	Validates that a user is in a group.

**Normal response codes:** 204

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

### 7.8.9.1. Request

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.

## 7.9. Credentials

Manage credentials.

Method	URI	Description
<b>POST</b>	/v3/credentials	Adds a credential.
<b>GET</b>	/v3/credentials{?page,per_page}	Lists credentials.
<b>GET</b>	/v3/credentials/{credential_id}	Shows details for a specified credential.
<b>PATCH</b>	/v3/credentials/{credential_id}	Updates a specified credential.
<b>DELETE</b>	/v3/credentials/{credential_id}	Deletes a specified credential.

## 7.9.1. Add credential

Method	URI	Description
POST	/v3/credentials	Adds a credential.

The following example shows how to create an EC2-style credential where the credential blob is a JSON-serialized dictionary that contains `access` and `secret`. This format is required when you specify the `ec2` type. To specify other credentials, such as `access_key`, simply change the type and contents of the data blob.

**Normal response codes:** 201

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

### 7.9.1.1. Request

#### Example 7.63. Add credential: JSON request

```
{
  "blob": {
    "access": "--access-key--",
    "secret": "--secret-key--"
  },
  "project_id": "--project-id--",
  "type": "ec2",
  "user_id": "--user--id--"
}
```

### 7.9.1.2. Response

#### Example 7.64. Add credential: JSON response

```
{
  "blob": {
    "access": "--access-key--",
    "secret": "--secret-key--"
  },
  "id": "--credential-id--",
  "links": {
    "self": "http://identity:35357/v3/credentials/--credential-id--"
  },
  "project_id": "--project-id--",
  "type": "ec2",
  "user_id": "--user--id--"
}
```

## 7.9.2. List credentials

Method	URI	Description
GET	/v3/credentials{?page,per_page}	Lists credentials.

**Normal response codes:** 200

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

### 7.9.2.1. 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
page	String <i>(Optional)</i>	Enables you to page through the list.
per_page	String <i>(Optional)</i>	Sets the page size for paging through the list. Default page size is 30.

This operation does not accept a request body.

### 7.9.2.2. Response

#### Example 7.65. List credentials: JSON response

```
[  
  {  
    "blob": {  
      "access": "--access-key--",  
      "secret": "--secret-key--"  
    },  
    "id": "--credential-id--",  
    "links": {  
      "self": "http://identity:35357/v3/credentials/--credential-id--"  
    },  
    "project_id": "--project-id--",  
    "type": "ec2",  
    "user_id": "--user--id--"  
  },  
  {  
    "blob": {  
      "access": "--access-key--",  
      "secret": "--secret-key--"  
    },  
  }]
```

```
        "id": "--credential-id--",
        "links": {
            "self": "http://identity:35357/v3/credentials/--credential-id--"
        },
        "project_id": "--project-id--",
        "type": "ec2",
        "user_id": "--user--id--"
    }
]
```

## 7.9.3. Show credential details

Method	URI	Description
GET	/v3/credentials/{credential_id}	Shows details for a specified credential.

**Normal response codes:** 200

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

### 7.9.3.1. Request

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.

### 7.9.3.2. Response

#### Example 7.66. Show credential details: JSON response

```
{
  "blob": {
    "access": "--access-key--",
    "secret": "--secret-key--"
  },
  "id": "--credential-id--",
  "links": {
    "self": "http://identity:35357/v3/credentials/--credential-id--"
  },
  "project_id": "--project-id--",
  "type": "ec2",
  "user_id": "--user--id--"
}
```

## 7.9.4. Update credential

Method	URI	Description
PATCH	/v3/credentials/{credential_id}	Updates a specified credential.

**Normal response codes:** 200

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

### 7.9.4.1. Request

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

Name	Type	Description
{credential_id}	Uuid	The credential ID.

#### Example 7.67. Update credential: JSON request

```
{
  "blob": {
    "access": "--access-key--",
    "secret": "--secret-key--"
  },
  "project_id": "--project-id--",
  "type": "ec2",
  "user_id": "--user--id--"
}
```

### 7.9.4.2. Response

#### Example 7.68. Update credential: JSON response

```
{
  "blob": {
    "access": "--access-key--",
    "secret": "--secret-key--"
  },
  "id": "--credential-id--",
  "links": {
    "self": "http://identity:35357/v3/credentials/--credential-id--"
  },
  "project_id": "--project-id--",
  "type": "ec2",
  "user_id": "--user--id--"
}
```

## 7.9.5. Delete credential

Method	URI	Description
<b>DELETE</b>	/v3/credentials/{credential_id}	Deletes a specified credential.

**Normal response codes:** 204

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

### 7.9.5.1. Request

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.

## 7.10. Roles

Manage roles.

Method	URI	Description
<b>POST</b>	/v3/roles	Adds a role.
<b>GET</b>	/v3/roles{?name,page,per_page}	Lists roles.
<b>PUT</b>	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Grants a role to a specified domain user.
<b>HEAD</b>	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Validates that a user has a role on a domain.
<b>DELETE</b>	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Revokes a role from a specified domain user.
<b>GET</b>	/v3/role_assignments{?group.id, role.id, scope.domain.id, scope.project.id, user.id, effective}	Lists role assignments.

## 7.10.1. Add role

Method	URI	Description
POST	/v3/roles	Adds a role.

**Normal response codes:** 201

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

### 7.10.1.1. Request

This table shows the header parameters for the add role request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 7.69. Add role: JSON request

```
{
  "role": {
    "name": "a role name"
  }
}
```

### 7.10.1.2. Response

#### Example 7.70. Add role: JSON response

```
{
  "id": "--role-id--",
  "links": {
    "self": "http://identity:35357/v3/roles/--role-id--"
  },
  "name": "a role name"
}
```

## 7.10.2. List roles

Method	URI	Description
GET	/v3/roles{?name,page,per_page}	Lists roles.

**Normal response codes:** 200

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

### 7.10.2.1. 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.
page	String <i>(Optional)</i>	Enables you to page through the list.
per_page	String <i>(Optional)</i>	Sets the page size for paging through the list. Default page size is 30.

This operation does not accept a request body.

### 7.10.2.2. Response

#### Example 7.71. List roles: JSON response

```
[  
  {  
    "id": "--role-id--",  
    "links": {  
      "self": "http://identity:35357/v3/roles/--role-id--"  
    },  
    "name": "a role name"  
  },  
  {  
    "id": "--role-id--",  
    "links": {  
      "self": "http://identity:35357/v3/roles/--role-id--"  
    },  
    "name": "a role name"  
  }  
]
```



## 7.10.3. Grant role to domain user

Method	URI	Description
PUT	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Grants a role to a specified domain 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)

### 7.10.3.1. Request

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.

## 7.10.4. 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:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), itemNotFound (404)

### 7.10.4.1. Request

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.

## 7.10.5. Revoke role from domain user

Method	URI	Description
<b>DELETE</b>	/v3/domains/{domain_id}/users/{user_id}/roles/{role_id}	Revokes a role from a specified domain 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)

### 7.10.5.1. Request

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.

## 7.10.6. 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 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 the specified user.

GET /role\_assignments?scope.project.id={project\_id} lists all role assignments for the specified 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 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.

**Normal response codes:** 200

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

### 7.10.6.1. Request

This operation does not accept a request body.

### 7.10.6.2. Response

#### Example 7.72. List role assignments: JSON response

```
GET /role_assignments?user.id={user_id}&scope.project.id={project_id}&effective
```

### Example 7.73. List role assignments: JSON response

```
{
  "role_assignments": [
    {
      "links": {
        "assignment": "http://identity:35357/v3/domains/--domain-id--/users/--user-id--/roles/--role-id--"
      },
      "role": {
        "id": "--role-id--"
      },
      "scope": {
        "domain": {
          "id": "--domain-id--"
        }
      },
      "user": {
        "id": "--user-id--"
      }
    },
    {
      "group": {
        "id": "--group-id--"
      },
      "links": {
        "assignment": "http://identity:35357/v3/projects/--project-id--/groups/--group-id--/roles/--role-id--"
      },
      "role": {
        "id": "--role-id--"
      },
      "scope": {
        "project": {
          "id": "--project-id--"
        }
      }
    }
  ],
  "links": {
    "self": "http://identity:35357/v3/role_assignments",
    "previous": null,
    "next": null
  }
}
```

### Example 7.74. List role assignments: JSON response

```
GET /role_assignments?user.id={user_id}&scope.project.id={project_id}&effective
```

### Example 7.75. List role assignments: JSON response

```
{
  "role_assignments": [
    {
      "links": {
        "assignment": "http://identity:35357/v3/domains/--domain-id--/users/--user-id--/roles/--role-id--"
      },
    }
  ]
}
```

```

    "role": {
      "id": "--role-id--"
    },
    "scope": {
      "domain": {
        "id": "--domain-id--"
      }
    },
    "user": {
      "id": "--user-id--"
    }
  },
  {
    "links": {
      "assignment": "http://identity:35357/v3/projects/--project-id--/groups/--group-id--/roles/--role-id--",
      "membership": "http://identity:35357/v3/groups/--group-id--/users/--user-id--"
    },
    "role": {
      "id": "--role-id--"
    },
    "scope": {
      "project": {
        "id": "--project-id--"
      }
    },
    "user": {
      "id": "--user-id--"
    }
  }
],
"links": {
  "self": "http://identity:35357/v3/role_assignments?effective",
  "previous": null,
  "next": null
}
}

```

## 7.11. Policies

Manage policies.

Method	URI	Description
<b>POST</b>	/v3/policies	Adds a policy.
<b>GET</b>	/v3/policies{?type,page,per_page}	Lists policies.
<b>GET</b>	/v3/policies/{policy_id}	Shows details for a specified policy.
<b>PATCH</b>	/v3/policies/{policy_id}	Updates a specified policy.
<b>DELETE</b>	/v3/policies/{policy_id}	Deletes a specified policy.

## 7.11.1. Add policy

Method	URI	Description
POST	/v3/policies	Adds a policy.

**Normal response codes:** 201

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

### 7.11.1.1. Request

This table shows the header parameters for the add policy request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 7.76. Add policy: JSON request

```
{
  "blob": {
    "access": "...",
    "secret": "..."
  },
  "project_id": "--project-id--",
  "type": "ec2",
  "user_id": "--user--id--"
}
```

### 7.11.1.2. Response

#### Example 7.77. Add policy: JSON response

```
{
  "blob": {
    "access": "...",
    "secret": "..."
  },
  "id": "--credential-id--",
  "links": {
    "self": "http://identity:35357/v3/credentials/--credential-id--"
  },
  "project_id": "--project-id--",
  "type": "ec2",
  "user_id": "--user--id--"
}
```

## 7.11.2. List policies

Method	URI	Description
GET	/v3/policies{?type,page,per_page}	Lists policies.

**Normal response codes:** 200

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

### 7.11.2.1. 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.
page	String <i>(Optional)</i>	Enables you to page through the list.
per_page	String <i>(Optional)</i>	Sets the page size for paging through the list. Default page size is 30.

This operation does not accept a request body.

### 7.11.2.2. Response

#### Example 7.78. List policies: JSON response

```
[  
  {  
    "blob": {  
      "access": "...",  
      "secret": "..."  
    },  
    "id": "--credential-id--",  
    "links": {  
      "self": "http://identity:35357/v3/credentials/--credential-id--"  
    },  
    "project_id": "--project-id--",  
    "type": "ec2",  
    "user_id": "--user--id--"  
  },  
  {  
    "blob": {  
      "access": "...",  
      "secret": "..."  
    },  
    "id": "--credential-id--",  
    "links": {  
      "self": "http://identity:35357/v3/credentials/--credential-id--"  
    },  
    "project_id": "--project-id--",  
    "type": "ec2",  
    "user_id": "--user--id--"  
  }]
```

```
        "access": "...",
        "secret": "..."
    },
    "id": "--credential-id--",
    "links": {
        "self": "http://identity:35357/v3/credentials/--credential-id--"
    },
    "project_id": "--project-id--",
    "type": "ec2",
    "user_id": "--user--id--"
}
]
```

## 7.11.3. Show policy details

Method	URI	Description
GET	/v3/policies/{policy_id}	Shows details for a specified policy.

**Normal response codes:** 200

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

### 7.11.3.1. 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.

### 7.11.3.2. Response

#### Example 7.79. Show policy details: JSON response

```
{
  "blob": {
    "access": "...",
    "secret": "..."
  },
  "id": "--credential-id--",
  "links": {
    "self": "http://identity:35357/v3/credentials/--credential-id--"
  },
  "project_id": "--project-id--",
  "type": "ec2",
  "user_id": "--user--id--"
}
```

## 7.11.4. Update policy

Method	URI	Description
PATCH	/v3/policies/{policy_id}	Updates a specified policy.

**Normal response codes:** 200

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

### 7.11.4.1. 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 7.80. Update policy: JSON request

```
{
  "blob": {
    "access": "...",
    "secret": "..."
  },
  "project_id": "--project-id--",
  "type": "ec2",
  "user_id": "--user--id--"
}
```

### 7.11.4.2. Response

#### Example 7.81. Update policy: JSON response

```
{
  "blob": {
    "access": "...",
    "secret": "..."
  },
  "id": "--credential-id--",
  "links": {
    "self": "http://identity:35357/v3/credentials/--credential-id--"
  },
  "project_id": "--project-id--",
  "type": "ec2",
  "user_id": "--user--id--"
}
```



## 7.11.5. Delete policy

Method	URI	Description
DELETE	/v3/policies/{policy_id}	Deletes a specified policy.

**Normal response codes:** 204

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

### 7.11.5.1. 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.

## 8. Identity API v3 extensions (STABLE)

Method	URI	Description
OS-OAUTH1 extension		
<b>POST</b>	/v3/OS-OAUTH1/consumers	Enables a user to create a consumer.
<b>GET</b>	/v3/OS-OAUTH1/consumers	Lists consumers.
<b>GET</b>	/v3/OS-OAUTH1/consumers/{consumer_id}	Shows information for a specified consumer.
<b>PATCH</b>	/v3/OS-OAUTH1/consumers/{consumer_id}	Updates the description for a specified consumer.
<b>DELETE</b>	/v3/OS-OAUTH1/consumers/{consumer_id}	Deletes a specified consumer.
<b>POST</b>	/v3/OS-OAUTH1/access_token	Enables a consumer to create an access token by exchanging a request token for an access token.
<b>GET</b>	/v3/OS-OAUTH1/users/{user_id}/access_tokens	Lists authorized access tokens.
<b>GET</b>	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}	Gets an authorized access token.
<b>DELETE</b>	/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.
<b>GET</b>	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}/roles	Lists associated roles for a specified access token.
<b>GET</b>	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}/roles/{role_id}	Gets information about a specified role for a specified access token.
<b>POST</b>	/v3/auth/tokens	Enables a consumer to get an Identity Service authentication token.

### 8.1. OS-OAUTH1 extension

Enable users to delegate roles to third-party consumers through the [OAuth 1.0a specification](#).

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 specified 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
<b>POST</b>	/v3/OS-OAUTH1/consumers	Enables a user to create a consumer.
<b>GET</b>	/v3/OS-OAUTH1/consumers	Lists consumers.
<b>GET</b>	/v3/OS-OAUTH1/consumers/{consumer_id}	Shows information for a specified consumer.
<b>PATCH</b>	/v3/OS-OAUTH1/consumers/{consumer_id}	Updates the description for a specified consumer.
<b>DELETE</b>	/v3/OS-OAUTH1/consumers/{consumer_id}	Deletes a specified consumer.
<b>POST</b>	/v3/OS-OAUTH1/access_token	Enables a consumer to create an access token by exchanging a request token for an access token.
<b>GET</b>	/v3/OS-OAUTH1/users/{user_id}/access_tokens	Lists authorized access tokens.
<b>GET</b>	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}	Gets an authorized access token.
<b>DELETE</b>	/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.
<b>GET</b>	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}/roles	Lists associated roles for a specified access token.
<b>GET</b>	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}/roles/{role_id}	Gets information about a specified role for a specified access token.
<b>POST</b>	/v3/auth/tokens	Enables a consumer to get an Identity Service authentication token.

## 8.1.1. Create consumer

Method	URI	Description
POST	/v3/OS-OAUTH1/consumers	Enables a user to create a consumer.

**Normal response codes:** 201

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

### 8.1.1.1. Request

#### Example 8.1. Create consumer: JSON request

```
{  
    "consumer": {  
        "description": "My consumer"  
    }  
}
```

### 8.1.1.2. Response

#### Example 8.2. Create consumer: JSON response

```
{  
    "consumer": {  
        "secret": "4c7832",  
        "description": "My consumer",  
        "id": "7fea2d",  
        "links": {  
            "self": "http://identity:35357/v3/OS-OAUTH1/consumers/7fea2d"  
        }  
    }  
}
```

## 8.1.2. List consumers

Method	URI	Description
GET	/v3/OS-OAUTH1/consumers	Lists consumers.

**Normal response codes:** 200

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

### 8.1.2.1. Request

This operation does not accept a request body.

### 8.1.2.2. Response

#### Example 8.3. 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"
    }
}
```

## 8.1.3. Show consumer

Method	URI	Description
GET	/v3/OS-OAUTH1/consumers/{consumer_id}	Shows information for a specified consumer.

**Normal response codes:** 200

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

### 8.1.3.1. Request

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

Name	Type	Description
{consumer_id}	Uuid	The ID of the consumer.

This operation does not accept a request body.

### 8.1.3.2. Response

#### Example 8.4. Show consumer: JSON response

```
{
  "consumer": {
    "id": "7fea2d",
    "description": "My consumer",
    "links": {
      "self": "http://identity:35357/v3/OS-OAUTH1/consumers/7fea2d"
    }
  }
}
```

## 8.1.4. Update consumer

Method	URI	Description
PATCH	/v3/OS-OAUTH1/consumers/{consumer_id}	Updates the description for a specified 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:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503)

### 8.1.4.1. Request

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

Name	Type	Description
{consumer_id}	Uuid	The ID of the consumer.

#### Example 8.5. Update consumer: JSON request

```
{
  "consumer": {
    "description": "My new consumer"
  }
}
```

### 8.1.4.2. Response

#### Example 8.6. Update consumer: JSON response

```
{
  "consumer": {
    "description": "My new consumer",
    "id": "7fea2d",
    "links": {
      "self": "http://identity:35357/v3/OS-OAUTH1/consumers/7fea2d"
    }
  }
}
```

## 8.1.5. Delete consumer

Method	URI	Description
<b>DELETE</b>	/v3/OS-OAUTH1/consumers/{consumer_id}	Deletes a specified 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:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503)

### 8.1.5.1. 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.

## 8.1.6. 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:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503)

### 8.1.6.1. Request

This operation does not accept a request body.

### 8.1.6.2. Response

#### Example 8.7. Create access token: application/txtresponse

```
oauth_token=accd36&oauth_token_secret=aa47da&oauth_expires_at=
2013-09-11T06:07:51.501805Z
```

## 8.1.7. 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:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503)

### 8.1.7.1. 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.

### 8.1.7.2. Response

#### Example 8.8. 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"
  }
}
```

## 8.1.8. 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:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503)

### 8.1.8.1. 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.

### 8.1.8.2. Response

#### Example 8.9. 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"
  }
}
```

## 8.1.9. Revoke access token

Method	URI	Description
<b>DELETE</b>	/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

### 8.1.9.1. 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.

## 8.1.10. 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 a specified access token.

See **GET /v3/roles** for an ([example](#)) of this response format.

**Normal response codes:** 200

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

### 8.1.10.1. 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.

## 8.1.11. Get role information for an access token

Method	URI	Description
GET	/v3/OS-OAUTH1/users/{user_id}/access_tokens/{access_token_id}/roles/{role_id}	Gets information about a specified role for a specified access token.

See **GET** /v3/roles/{role\_id} for an [example](#) of this response format.

**Normal response codes:** 200

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

### 8.1.11.1. Request

This table shows the URI parameters for the get role information 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.

### 8.1.11.2. Response

This operation does not return a response body.

## 8.1.12. 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.

Various [examples](#) of OpenStack token responses.

Example OAuth-specific object in a token:

```
"OS-OAUTH1" : {  
    "access_token_id": "cce0b8be7",  
}
```

**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)

### 8.1.12.1. Request

This operation does not accept a request body.

### 8.1.12.2. Response

This operation does not return a response body.

## 9. Identity API v2.0 (STABLE)

Get an authentication token that permits access to the OpenStack services REST API.

### 9.1. API versions

Method	URI	Description
<b>GET</b>	/	Lists information about all Identity API versions.
<b>GET</b>	/v2.0	Shows details for the Identity API v2.0.

## 9.1.1. List versions

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

**Normal response codes:** 200

**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. List versions: JSON response

```
{
  "versions": [
    {
      "values": [
        {
          "id": "v3.0",
          "links": [
            {
              "href": "http://192.168.122.176:5000/v3/",
              "rel": "self"
            }
          ],
          "media-types": [
            {
              "base": "application/json",
              "type": "application/vnd.openstack.identity-v3+json"
            },
            {
              "base": "application/xml",
              "type": "application/vnd.openstack.identity-v3+xml"
            }
          ],
          "status": "stable",
          "updated": "2013-03-06T00:00:00Z"
        },
        {
          "id": "v2.0",
          "links": [
            {
              "href": "http://192.168.122.176:5000/v2.0/",
              "rel": "self"
            },
            {
              "href": "http://docs.openstack.org/",
              "rel": "describedby",
              "type": "text/html"
            }
          ]
        }
      ]
    }
  ]
}
```

```
        }
    ],
    "media-types": [
        {
            "base": "application/json",
            "type": "application/vnd.openstack.identity-v2.0+json"
        },
        {
            "base": "application/xml",
            "type": "application/vnd.openstack.identity-v2.0+xml"
        }
    ],
    "status": "stable",
    "updated": "2014-04-17T00:00:00Z"
}
]
}
```

### Example 9.2. List versions: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<versions xmlns="http://docs.openstack.org/identity/api/v2.0">
    <version status="stable" updated="2013-03-06T00:00:00Z" id="v3.0">
        <media-types>
            <media-type base="application/json" type="application/vnd.openstack.identity-v3+json"/>
            <media-type base="application/xml" type="application/vnd.openstack.identity-v3+xml"/>
        </media-types>
        <links>
            <link href="http://192.168.122.176:5000/v3/" rel="self"/>
        </links>
    </version>
    <version status="stable" updated="2014-04-17T00:00:00Z" id="v2.0">
        <media-types>
            <media-type base="application/json" type="application/vnd.openstack.identity-v2.0+json"/>
            <media-type base="application/xml" type="application/vnd.openstack.identity-v2.0+xml"/>
        </media-types>
        <links>
            <link href="http://192.168.122.176:5000/v2.0/" rel="self"/>
            <link href="http://docs.openstack.org/" type="text/html" rel="describedby"/>
        </links>
        <link href="http://192.168.122.176:5000/v2.0/" rel="self"/>
        <link href="http://docs.openstack.org/" type="text/html" rel="describedby"/>
    </version>
</versions>
```

This operation does not return a response body.

## 9.1.2. 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)

### 9.1.2.1. Request

This operation does not accept a request body.

### 9.1.2.2. Response

#### Example 9.3. Get version information: JSON response

```
{
  "version": {
    "status": "stable",
    "updated": "2013-03-06T00:00:00Z",
    "media-types": [
      {
        "base": "application/json",
        "type": "application/vnd.openstack.identity-v3+json"
      },
      {
        "base": "application/xml",
        "type": "application/vnd.openstack.identity-v3+xml"
      }
    ],
    "id": "v3.0",
    "links": [
      {
        "href": "http://23.253.228.211:35357/v3/",
        "rel": "self"
      }
    ]
  }
}
```

#### Example 9.4. Get version information: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<version xmlns="http://docs.openstack.org/identity/api/v3"
  status="stable" updated="2013-03-06T00:00:00Z" id="v3.0">
  <media-types>
    <media-type base="application/json"
      type="application/vnd.openstack.identity-v3+json"/>
    <media-type base="application/xml"
      type="application/vnd.openstack.identity-v3+xml"/>
  </media-types>
  <links>
```

```
        <link href="http://23.253.228.211:35357/v3/" rel="self"/>
    </links>
</version>
```

## 9.2. Extensions

Method	URI	Description
<b>GET</b>	/v2.0/extensions	Lists available extensions.
<b>GET</b>	/v2.0/extensions/{alias}	Gets detailed information for a specified extension.

## 9.2.1. List extensions

Method	URI	Description
GET	/v2.0/extensions	Lists available extensions.

**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)

### 9.2.1.1. Request

This operation does not accept a request body.

### 9.2.1.2. Response

#### Example 9.5. List extensions: JSON response

```
{
  "extensions": {
    "values": [
      {
        "updated": "2013-07-07T12:00:0-00:00",
        "name": "OpenStack S3 API",
        "links": [
          {
            "href": "https://github.com/openstack/identity-api",
            "type": "text/html",
            "rel": "describedby"
          }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/s3tokens/v1.0",
        "alias": "s3tokens",
        "description": "OpenStack S3 API."
      },
      {
        "updated": "2013-07-23T12:00:0-00:00",
        "name": "OpenStack Keystone Endpoint Filter API",
        "links": [
          {
            "href": "https://github.com/openstack/identity-api/blob/master/openstack-identity-api/v3/src/markdown/identity-api-v3-os-ep-filter-ext.md",
            "type": "text/html",
            "rel": "describedby"
          }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-EP-FILTER/v1.0",
        "alias": "OS-EP-FILTER",
        "description": "OpenStack Keystone Endpoint Filter API."
      },
      {
        "updated": "2013-12-17T12:00:0-00:00",
        "name": "OpenStack Federation APIs",
        "links": [
          {
            "href": "https://github.com/openstack/identity-api/blob/master/openstack-identity-api/v3/src/markdown/identity-api-v3-federation-ext.md",
            "type": "text/html",
            "rel": "describedby"
          }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/FEDERATION/v1.0",
        "alias": "OS-FEDERATION",
        "description": "OpenStack Federation APIs."
      }
    ]
  }
}
```

```
        "links": [
            {
                "href": "https://github.com/openstack/identity-api",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-FEDERATION/v1.0",
        "alias": "OS-FEDERATION",
        "description": "OpenStack Identity Providers Mechanism."
    },
    {
        "updated": "2013-07-11T17:14:00-00:00",
        "name": "OpenStack Keystone Admin",
        "links": [
            {
                "href": "https://github.com/openstack/identity-api",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-KSADM/v1.0",
        "alias": "OS-KSADM",
        "description": "OpenStack extensions to Keystone v2.0 API enabling Administrative Operations."
    },
    {
        "updated": "2014-01-20T12:00:0-00:00",
        "name": "OpenStack Simple Certificate API",
        "links": [
            {
                "href": "https://github.com/openstack/identity-api",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-SIMPLE-CERT/v1.0",
        "alias": "OS-SIMPLE-CERT",
        "description": "OpenStack simple certificate retrieval extension"
    },
    {
        "updated": "2013-07-07T12:00:0-00:00",
        "name": "OpenStack EC2 API",
        "links": [
            {
                "href": "https://github.com/openstack/identity-api",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-EC2/v1.0",
        "alias": "OS-EC2",
        "description": "OpenStack EC2 Credentials backend."
    }
]
```

```
    }  
}
```

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

Name	Type	Description
next	AnyURI <i>(Optional)</i>	Moves to the next item in the list.
previous	AnyURI <i>(Optional)</i>	Moves to the previous item in the list.

### Example 9.6. List extensions: XML response

```
<?xml version="1.0" encoding="UTF-8"?>  
<extensions xmlns="http://docs.openstack.org/common/api/v1.0"  
           xmlns:atom="http://www.w3.org/2005/Atom"/>
```

## 9.2.2. Get extension details

Method	URI	Description
GET	/v2.0/extensions/{alias}	Gets detailed information for a specified extension.

Specify the extension alias in the URI.

**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)

### 9.2.2.1. Request

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

Name	Type	Description
{alias}	String	The extension name.

This operation does not accept a request body.

### 9.2.2.2. Response

#### Example 9.7. Get extension details: JSON response

```
{
  "extension": {
    "updated": "2013-07-07T12:00:0-00:00",
    "name": "OpenStack S3 API",
    "links": [
      {
        "href": "https://github.com/openstack/identity-api",
        "type": "text/html",
        "rel": "describedby"
      }
    ],
    "namespace": "http://docs.openstack.org/identity/api/ext/s3tokens/v1.
0",
    "alias": "s3tokens",
    "description": "OpenStack S3 API."
  }
}
```

#### Example 9.8. Get extension details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<extension xmlns="http://docs.openstack.org/common/api/v1.0"
           xmlns:atom="http://www.w3.org/2005/Atom"
           name="User Metadata Extension"
           namespace="http://docs.rackspacecloud.com/identity/api/ext/meta/v2.0"
           alias="RS-META" updated="2011-01-12T11:22:33-06:00">
  <description>Allows associating arbitrary metadata with a user.</description>
```

```
<atom:link rel="describedby" type="application/pdf"
    href="http://docs.rackspacecloud.com/identity/api/ext/identity-
meta-20111201.pdf"/>
<atom:link rel="describedby" type="application/vnd.sun.wadl+xml"
    href="http://docs.rackspacecloud.com/identity/api/ext/identity-meta.
wadl"
    />
</extension>
```

## 9.3. Tokens

Method	URI	Description
<b>POST</b>	/v2.0/tokens	Authenticates and generates a token.
<b>GET</b>	/v2.0/tenants{?marker,limit}	Lists tenants to which the specified token has access.

## 9.3.1. Authenticate

Method	URI	Description
POST	/v2.0/tokens	Authenticates and generates a token.

The Identity API is a ReSTful web service. It is the entry point to all service APIs. To access the Identity API, you must know its URL.

Each ReST request against Identity requires the X-Auth-Token header. Clients obtain this token, along with the URL to other service APIs, by first authenticating against Identity with valid credentials.

To authenticate, you must provide either a user ID and password or a token.

If the authentication token has expired, a 401 response code is returned.

If the token specified in the request has expired, this call returns a 404 response code.

Identity treats expired tokens as invalid tokens.

The deployment determines how long expired tokens are stored.

**Normal response codes:** 200, 203

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

### 9.3.1.1. Request

#### Example 9.9. Authenticate with user name and password credentials: JSON request

```
{
  "auth": {
    "tenantName": "demo",
    "passwordCredentials": {
      "username": "demo",
      "password": "devstack"
    }
  }
}
```

This table shows the body parameters for the authenticate request:

Name	Type	Description
tenantName	String (Optional)	The tenant name. Both the tenantId and tenantName attributes are optional, but should not be specified together. If both attributes are specified, the server responds with a 400 Bad Request.
tenantId	UUID (Optional)	The tenant ID. Both the tenantId and tenantName attributes are optional, but should not be specified together. If both attributes are specified, the server responds with a 400 Bad Request.
passwordCredentials	String (Optional)	A passwordCredentials 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. If you do not provide a password credentials, you must provide a token.
password	String <i>(Optional)</i>	The password of the user. Required if you include the passwordCredentials object. If you do not provide a password credentials, you must provide a token.
token	String <i>(Optional)</i>	A token object. Required if you do not provide password credentials.
id	String <i>(Optional)</i>	The token ID. This is a required field in the token object.

### Example 9.10. 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="devstack"/>
</auth>
```

### Example 9.11. Authenticate with token: JSON request

```
{
  "auth": {
    "tenantName": "demo",
    "token": {
      "id": "cbc36478b0bd8e67e89469c7749d4127"
    }
  }
}
```

### Example 9.12. Authenticate with token: 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">
    <token id="cbc36478b0bd8e67e89469c7749d4127"/>
</auth>
```

## 9.3.1.2. Response

### Example 9.13. 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": "admin"
      }
    }
  }
}
```

```
        "name": "demo"
    },
},
"serviceCatalog": [
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8774/v2/
fc394f2ab2df4114bde39905f800dc57",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8774/v2/
fc394f2ab2df4114bde39905f800dc57",
            "id": "2dad48f09e2a447a9bf852bcd93548ef",
            "publicURL": "http://23.253.72.207:8774/v2/
fc394f2ab2df4114bde39905f800dc57"
        }
    ],
    "endpoints_links": [],
    "type": "compute",
    "name": "nova"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:9696/",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:9696/",
            "id": "97c526db8d7a4c88bbb8d68db1bdcdb8",
            "publicURL": "http://23.253.72.207:9696/"
        }
    ],
    "endpoints_links": [],
    "type": "network",
    "name": "neutron"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8776/v2/
fc394f2ab2df4114bde39905f800dc57",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8776/v2/
fc394f2ab2df4114bde39905f800dc57",
            "id": "93f86dfcbba143a39a33d0c2cd424870",
            "publicURL": "http://23.253.72.207:8776/v2/
fc394f2ab2df4114bde39905f800dc57"
        }
    ],
    "endpoints_links": [],
    "type": "volumev2",
    "name": "cinder"
},
{
    "endpoints": [
        {
            "adminURL": "http://23.253.72.207:8774/v3",
            "region": "RegionOne",
            "internalURL": "http://23.253.72.207:8774/v3",
            "id": "3eb274b12b1d47b2abc536038d87339e",
            "publicURL": "http://23.253.72.207:8774/v3"
        }
    ]
}
```

```
        }
    ],
    "endpoints_links": [],
    "type": "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": []
}
```

```
        "publicURL": "http://23.253.72.207:8773/services/
Cloud"
    }
],
"endpoints_links": [],
"type": "ec2",
"name": "ec2"
},
{
"endpoints": [
{
"adminURL": "http://23.253.72.207:8080",
"region": "RegionOne",
"internalURL": "http://23.253.72.207:8080/v1/
AUTH_fc394f2ab2df4114bde39905f800dc57",
"id": "16b76b5e5b7d48039a6e4cc3129545f3",
"publicURL": "http://23.253.72.207:8080/v1/
AUTH_fc394f2ab2df4114bde39905f800dc57"
}
],
"endpoints_links": [],
"type": "object-store",
"name": "swift"
},
{
"endpoints": [
{
"adminURL": "http://23.253.72.207:35357/v2.0",
"region": "RegionOne",
"internalURL": "http://23.253.72.207:5000/v2.0",
"id": "26af053673df4ef3a2340c4239e21ea2",
"publicURL": "http://23.253.72.207:5000/v2.0"
}
],
"endpoints_links": [],
"type": "identity",
"name": "keystone"
}
],
"user": {
"username": "demo",
"roles_links": [],
"id": "9a6590b2ab024747bc2167c4e064d00d",
"roles": [
{
"name": "Member"
},
{
"name": "anotherrole"
}
],
"name": "demo"
},
"metadata": {
"is_admin": 0,
"roles": [
"7598ac3c634d4c3da4b9126a5f67ca2b",
"f95c0ab82d6045d9805033ee1fbc80d4"
]
}
}
```

```

    }
}
```

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	String <i>(Required)</i>	A timestamp that indicates when the token was issued.
expires	String <i>(Required)</i>	A timestamp that indicates when the token 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 <i>(Required)</i>	A user object, which shows the username, roles_links, id, roles, and name.
metadata	String <i>(Required)</i>	A metadata object.

### Example 9.14. 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-dddd">
        <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"
                />
            </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"
                />
            </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>
</access>
```

## 9.3.2. List tenants

Method	URI	Description
GET	/v2.0/tenants{?marker,limit}	Lists tenants to which the specified token has access.

**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)

### 9.3.2.1. 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 9.15. 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
```

#### Example 9.16. List tenants: HTTP/XML request

```
GET /v2.0/tenants HTTP/1.1
Host: identity.api.openstack.org
Content-Type: application/xml
X-Auth-Token: fa8426a0-8eaf-4d22-8e13-7c1b16a9370c
Accept: application/xml
```

This operation does not accept a request body.

### 9.3.2.2. Response

#### Example 9.17. 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
    }
  ]
}
```

```
        "enabled": true
    }
],
"tenants_links": []
}
```

### Example 9.18. List tenants: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<tenants xmlns="http://docs.openstack.org/identity/api/v2.0">
    <tenant enabled="true" id="1234" name="ACME Corp">
        <description>A description...</description>
    </tenant>
    <tenant enabled="true" id="3645" name="Iron Works">
        <description>A description...</description>
    </tenant>
</tenants>
```

# 10. Identity admin API v2.0 (STABLE)

Get an authentication token that permits access to the Compute API.

Method	URI	Description
Versions		
<b>GET</b>	/v2.0	Gets detailed information about a specified version of the Identity API.
Extensions		
<b>GET</b>	/v2.0/extensions	Lists available extensions.
<b>GET</b>	/v2.0/extensions/{alias}	Gets detailed information for a specified extension.
Tokens		
<b>POST</b>	/v2.0/tokens	Authenticates and generates a token.
<b>GET</b>	/v2.0/tokens/{tokenId}{?belongsTo}	Validates a token and confirms that it belongs to a specified tenant.
<b>HEAD</b>	/v2.0/tokens/{tokenId}{?belongsTo}	Validates a token and confirms that it belongs to a specified tenant, for performance.
Users		
<b>POST</b>	/v2.0/users	Adds a user.
<b>PUT</b>	/v2.0/users/{userId}	Updates a user.
<b>DELETE</b>	/v2.0/users/{userId}	Deletes a user.
<b>GET</b>	/v2.0/users/{name}{?name}	Gets detailed information about a specified user by user name.
<b>GET</b>	/v2.0/users/{user_id}	Gets detailed information about a specified user by user ID.
<b>GET</b>	/v2.0/users/{user_id}/roles	Lists global roles for a specified user. Excludes tenant roles.
Tenants		
<b>GET</b>	/v2.0/tenants{?marker,limit}	Lists all tenants.
<b>GET</b>	/v2.0/tenants{?marker,limit,name}	Gets detailed information about a specified tenant by name.
<b>GET</b>	/v2.0/tenants/{tenantId}	Gets detailed information about a specified tenant by ID.
<b>GET</b>	/v2.0/tenants/{tenantId}/users/{userId}/roles	Lists roles for a specified user on a specified tenant. Excludes global roles.

## 10.1. Versions

Method	URI	Description
<b>GET</b>	/v2.0	Gets detailed information about a specified version of the Identity API.

## 10.1.1. Get version details

Method	URI	Description
GET	/v2.0	Gets detailed information about a specified version of the Identity API.

**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)

### 10.1.1.1. Request

This operation does not accept a request body.

### 10.1.1.2. Response

#### Example 10.1. Get version information: JSON response

```
{
    "version": {
        "status": "stable",
        "updated": "2014-04-17T00:00:00Z",
        "media-types": [
            {
                "base": "application/json",
                "type": "application/vnd.openstack.identity-v2.0+json"
            },
            {
                "base": "application/xml",
                "type": "application/vnd.openstack.identity-v2.0+xml"
            }
        ],
        "id": "v2.0",
        "links": [
            {
                "href": "http://23.253.228.211:5000/v2.0/",
                "rel": "self"
            },
            {
                "href": "http://docs.openstack.org/api/openstack-identity-service/2.0/content/",
                "type": "text/html",
                "rel": "describedby"
            },
            {
                "href": "http://docs.openstack.org/api/openstack-identity-service/2.0/identity-dev-guide-2.0.pdf",
                "type": "application/pdf",
                "rel": "describedby"
            }
        ]
    }
}
```

This table shows the body parameters for the get version details response:

Name	Type	Description
location	AnyURI (Required)	

### Example 10.2. Get version information: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<version xmlns="http://docs.openstack.org/identity/api/v2.0"
    status="stable" updated="2013-03-06T00:00:00Z" id="v2.0">
    <media-types>
        <media-type base="application/json"
            type="application/vnd.openstack.identity-v2.0+json"/>
        <media-type base="application/xml"
            type="application/vnd.openstack.identity-v2.0+xml"/>
    </media-types>
    <links>
        <link href="http://localhost:5000/v2.0/" rel="self"/>
        <link
            href="http://docs.openstack.org/api/openstack-identity-service/2.0/content/"
            type="text/html" rel="describedby"/>
        <link
            href="http://docs.openstack.org/api/openstack-identity-service/2.0/identity-dev-guide-2.0.pdf"
            type="application/pdf" rel="describedby"/>
    </links>
</version>
```

## 10.2. Extensions

Method	URI	Description
GET	/v2.0/extensions	Lists available extensions.
GET	/v2.0/extensions/{alias}	Gets detailed information for a specified extension.

## 10.2.1. List extensions

Method	URI	Description
GET	/v2.0/extensions	Lists available extensions.

**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 operation does not accept a request body.

### 10.2.1.2. Response

#### Example 10.3. List extensions: JSON response

```
{
  "extensions": [
    {
      "values": [
        {
          "updated": "2013-07-07T12:00:0-00:00",
          "name": "OpenStack S3 API",
          "links": [
            {
              "href": "https://github.com/openstack/identity-api",
              "type": "text/html",
              "rel": "describedby"
            }
          ],
          "namespace": "http://docs.openstack.org/identity/api/ext/s3tokens/v1.0",
          "alias": "s3tokens",
          "description": "OpenStack S3 API."
        },
        {
          "updated": "2013-07-23T12:00:0-00:00",
          "name": "OpenStack Keystone Endpoint Filter API",
          "links": [
            {
              "href": "https://github.com/openstack/identity-api/blob/master/openstack-identity-api/v3/src/markdown/identity-api-v3-os-ep-filter-ext.md",
              "type": "text/html",
              "rel": "describedby"
            }
          ],
          "namespace": "http://docs.openstack.org/identity/api/ext/OS-EP-FILTER/v1.0",
          "alias": "OS-EP-FILTER",
          "description": "OpenStack Keystone Endpoint Filter API."
        },
        {
          "updated": "2013-12-17T12:00:0-00:00",
          "name": "OpenStack Federation APIs",
          "links": [
            {
              "href": "https://github.com/openstack/identity-api/blob/master/openstack-identity-api/v3/src/markdown/identity-api-v3-federation-ext.md",
              "type": "text/html",
              "rel": "describedby"
            }
          ],
          "namespace": "http://docs.openstack.org/identity/api/ext/FEDERATION/v1.0",
          "alias": "OS-FEDERATION",
          "description": "OpenStack Federation APIs."
        }
      ]
    }
  ]
}
```

```
        "links": [
            {
                "href": "https://github.com/openstack/identity-api",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-FEDERATION/v1.0",
        "alias": "OS-FEDERATION",
        "description": "OpenStack Identity Providers Mechanism."
    },
    {
        "updated": "2013-07-11T17:14:00-00:00",
        "name": "OpenStack Keystone Admin",
        "links": [
            {
                "href": "https://github.com/openstack/identity-api",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-KSADM/v1.0",
        "alias": "OS-KSADM",
        "description": "OpenStack extensions to Keystone v2.0 API enabling Administrative Operations."
    },
    {
        "updated": "2014-01-20T12:00:0-00:00",
        "name": "OpenStack Simple Certificate API",
        "links": [
            {
                "href": "https://github.com/openstack/identity-api",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-SIMPLE-CERT/v1.0",
        "alias": "OS-SIMPLE-CERT",
        "description": "OpenStack simple certificate retrieval extension"
    },
    {
        "updated": "2013-07-07T12:00:0-00:00",
        "name": "OpenStack EC2 API",
        "links": [
            {
                "href": "https://github.com/openstack/identity-api",
                "type": "text/html",
                "rel": "describedby"
            }
        ],
        "namespace": "http://docs.openstack.org/identity/api/ext/OS-EC2/v1.0",
        "alias": "OS-EC2",
        "description": "OpenStack EC2 Credentials backend."
    }
]
```

```
    }  
}
```

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

Name	Type	Description
next	AnyURI <i>(Optional)</i>	Moves to the next item in the list.
previous	AnyURI <i>(Optional)</i>	Moves to the previous item in the list.

#### Example 10.4. List extensions: XML response

```
<?xml version="1.0" encoding="UTF-8"?>  
<extensions xmlns="http://docs.openstack.org/common/api/v1.0"  
           xmlns:atom="http://www.w3.org/2005/Atom"/>
```

## 10.2.2. Get extension details

Method	URI	Description
GET	/v2.0/extensions/{alias}	Gets detailed information for a specified extension.

Specify the extension alias in the URI.

**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)

### 10.2.2.1. Request

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

Name	Type	Description
{alias}	String	The extension name.

This operation does not accept a request body.

### 10.2.2.2. Response

#### Example 10.5. Get extension details: JSON response

```
{
  "extension": {
    "updated": "2013-07-07T12:00:0-00:00",
    "name": "OpenStack S3 API",
    "links": [
      {
        "href": "https://github.com/openstack/identity-api",
        "type": "text/html",
        "rel": "describedby"
      }
    ],
    "namespace": "http://docs.openstack.org/identity/api/ext/s3tokens/v1.
0",
    "alias": "s3tokens",
    "description": "OpenStack S3 API."
  }
}
```

#### Example 10.6. Get extension details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<extension xmlns="http://docs.openstack.org/common/api/v1.0"
           xmlns:atom="http://www.w3.org/2005/Atom"
           name="User Metadata Extension"
           namespace="http://docs.rackspacecloud.com/identity/api/ext/meta/v2.0"
           alias="RS-META" updated="2011-01-12T11:22:33-06:00">
  <description>Allows associating arbitrary metadata with a user.</description>
```

```
<atom:link rel="describedby" type="application/pdf"
    href="http://docs.rackspacecloud.com/identity/api/ext/identity-
meta-20111201.pdf"/>
<atom:link rel="describedby" type="application/vnd.sun.wadl+xml"
    href="http://docs.rackspacecloud.com/identity/api/ext/identity-meta.
wadl"
    />
</extension>
```

## 10.3. Tokens

Method	URI	Description
<b>POST</b>	/v2.0/tokens	Authenticates and generates a token.
<b>GET</b>	/v2.0/tokens/{tokenId}{?belongsTo}	Validates a token and confirms that it belongs to a specified tenant.
<b>HEAD</b>	/v2.0/tokens/{tokenId}{?belongsTo}	Validates a token and confirms that it belongs to a specified tenant, for performance.

## 10.3.1. Authenticate for Admin API

Method	URI	Description
POST	/v2.0/tokens	Authenticates and generates a token.

Client authentication is provided through a ReST interface by using the POST method with v2.0/tokens supplied 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, along with the URL to other service APIs, by first authenticating against Identity with valid credentials.

If the authentication token has expired, a 401 response code is returned.

If the token specified in the request body has expired, this call returns a 404 response code.

Identity treats expired tokens as invalid tokens.

The deployment determines how long expired tokens are stored.

**Normal response codes:** 200, 203

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

### 10.3.1.1. Request

#### Example 10.7. Authenticate with credentials: JSON request

```
{
  "auth": {
    "tenantName": "admin",
    "passwordCredentials": {
      "username": "admin",
      "password": "devstack"
    }
  }
}
```

This table shows the body parameters for the authenticate for admin api request:

Name	Type	Description
tenantName	String (Optional)	The tenant name. Both the tenantId and tenantName attributes are optional, but should not be specified together. If both attributes are specified, the server responds with a 400 Bad Request.
tenantId	UUID (Optional)	The tenant ID. Both the tenantId and tenantName attributes are optional, but should not be specified together. If both attributes are specified, the server responds with a 400 Bad Request.
passwordCredentials	String (Optional)	A passwordCredentials object. To authenticate, you must provide either a user ID and password or a token.

Name	Type	Description
username	String (Optional)	The user name. Required if you include the passwordCredentials object. If you do not provide a password credentials, you must provide a token.
password	String (Optional)	The password of the user. Required if you include the passwordCredentials object. If you do not provide a password credentials, you must provide a token.
token	String (Optional)	A token object. Required if you do not provide password credentials.
id	String (Optional)	The token ID. This is a required field in the token object.

### Example 10.8. Authenticate with 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="admin">
    <passwordCredentials username="admin" password="devstack"/>
</auth>
```

### Example 10.9. Authenticate with token: JSON request

```
{
  "auth": {
    "tenantName": "demo",
    "token": {
      "id": "cbc36478b0bd8e67e89469c7749d4127"
    }
  }
}
```

### Example 10.10. Authenticate with token: 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">
    <token id="cbc36478b0bd8e67e89469c7749d4127"/>
</auth>
```

## 10.3.1.2. Response

### Example 10.11. 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": "93f86dfcbbal43a39a33d0c2cd424870",
                    "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"
    ]
}
}
```

This table shows the body parameters for the authenticate for admin api response:

Name	Type	Description
access	String <i>(Required)</i>	An access object.
token	String <i>(Required)</i>	A token object.
issued_at	String <i>(Required)</i>	A timestamp that indicates when the token was issued.
expires	String <i>(Required)</i>	A timestamp that indicates when the token expires.
id	String <i>(Required)</i>	The authentication token. In the example, the token is <code>my_id</code> .
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 <code>adminURL</code> , <code>region</code> , <code>internalURL</code> , <code>id</code> , and <code>publicURL</code> 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 <i>(Required)</i>	A user object, which shows the <code>username</code> , <code>roles_links</code> , <code>id</code> , <code>roles</code> , and <code>name</code> .
metadata	String <i>(Required)</i>	A metadata object.

### Example 10.12. Authenticate with 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="admin_id">
```

```
<tenant enabled="true" name="admin"
        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"
        />
    </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="97c526db8d7a4c88bbb8d68db1bdcdcb8"
        />
    </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="93f86dfcbb143a39a33d0c2cd424870"
        />
    </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="admin" id="9a6590b2ab024747bc2167c4e064d00d"
      name="admin">
    <roles_links/>
    <role name="Member" />
```

```
        <role name="anotherrole"/>
    </user>
    <metadata is_admin="0">
        <roles>
            <role>7598ac3c634d4c3da4b9126a5f67ca2b</role>
            <role>f95c0ab82d6045d9805033ee1fbc80d4</role>
        </roles>
    </metadata>
</access>
```

## 10.3.2. Validate token

Method	URI	Description
GET	/v2.0/tokens/{tokenId}{?belongsTo}	Validates a token and confirms that it belongs to a specified tenant.

Returns the permissions relevant to a particular client. Valid tokens are in the /tokens/{tokenId} path. A user should expect an itemNotFound (404) fault for an token that is not valid.

**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)

### 10.3.2.1. Request

This table shows the URI parameters for the validate token request:

Name	Type	Description
{tokenId}	UUID	Required. The token ID.

This operation does not accept a request body.

### 10.3.2.2. Response

#### Example 10.13. 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": [
        ...
      ]
    }
  }
}
```

```
        ]
    }
}
```

### Example 10.14. Validate token: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<access xmlns="http://docs.openstack.org/identity/api/v2.0">
    <token id="ab48a9efdfedb23ty3494"
        expires="2010-11-01T03:32:15-05:00">
        <tenant id="456" name="My Project"/>
    </token>
    <user id="123" username="jqsmith">
        <roles
            xmlns="http://docs.openstack.org/identity/api/v2.0">
            <role id="123" name="Admin" tenantId="one"/>
            <role id="234" name="object-store:admin"
                tenantId="1"/>
        </roles>
    </user>
</access>
```

### 10.3.3. Validate token (admin)

Method	URI	Description
<b>HEAD</b>	/v2.0/tokens/{tokenId}{?belongsTo}	Validates a token and confirms that it belongs to a specified tenant, for performance.

**Normal response codes:** 200, 203, 204

**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 URI parameters for the validate token (admin) request:

Name	Type	Description
{tokenId}	UUID	Required. The token ID.

This operation does not accept a request body.

## 10.4. Users

Method	URI	Description
<b>POST</b>	/v2.0/users	Adds a user.
<b>PUT</b>	/v2.0/users/{userId}	Updates a user.
<b>DELETE</b>	/v2.0/users/{userId}	Deletes a user.
<b>GET</b>	/v2.0/users/{name}{?name}	Gets detailed information about a specified user by user name.
<b>GET</b>	/v2.0/users/{user_id}	Gets detailed information about a specified user by user ID.
<b>GET</b>	/v2.0/users/{user_id}/roles	Lists global roles for a specified user. Excludes tenant roles.

## 10.4.1. Add user

Method	URI	Description
POST	/v2.0/users	Adds 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)

### 10.4.1.1. Request

This table shows the header parameters for the add user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 10.15. Add user: JSON request

```
{
  "user": {
    "id": "u1000",
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

This table shows the body parameters for the add user request:

Name	Type	Description
id	String <i>(Required)</i>	The user ID.
name	String <i>(Required)</i>	The user name.
email	String <i>(Required)</i>	The user email.
enabled	Bool <i>(Required)</i>	Indicates whether the user is enabled (true) or disabled (false).

#### Example 10.16. Add user: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<user xmlns="http://docs.openstack.org/identity/api/v2.0"
      enabled="true" email="john.smith@example.org" name="jqsmith"
      id="u1000"/>
```

### 10.4.1.2. Response

#### Example 10.17. Add user: JSON response

```
{  
    "user": {  
        "id": "u1000",  
        "name": "jqsmith",  
        "email": "john.smith@example.org",  
        "enabled": true  
    }  
}
```

This table shows the body parameters for the add user response:

Name	Type	Description
id	String <i>(Required)</i>	The user ID.
name	String <i>(Required)</i>	The user name.
email	String <i>(Required)</i>	The user email.
enabled	Bool <i>(Required)</i>	Indicates whether the user is enabled (true) or disabled (false).

#### Example 10.18. Add user: XML response

```
<?xml version="1.0" encoding="UTF-8"?>  
<user xmlns="http://docs.openstack.org/identity/api/v2.0"  
      enabled="true" email="john.smith@example.org" name="jqsmith"  
      id="u1000"/>
```

This operation does not return a response body.

## 10.4.2. 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)

### 10.4.2.1. Request

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

Name	Type	Description
{userId}	String	The ID of the user for which you want to perform the request.

#### Example 10.19. Update user: JSON request

```
{
  "user": {
    "id": "u1000",
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

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

Name	Type	Description
id	String <i>(Required)</i>	The user ID.
name	String <i>(Required)</i>	The user name.
email	String <i>(Required)</i>	The user email.
enabled	Bool <i>(Required)</i>	Indicates whether the user is enabled (true) or disabled (false).

#### Example 10.20. Update user: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<user xmlns="http://docs.openstack.org/identity/api/v2.0"
      enabled="true" email="john.smith@example.org" name="jqsmith"
      id="u1000"/>
```

### 10.4.2.2. Response

#### Example 10.21. Update user: JSON response

```
{
```

```
"user": {  
    "id": "u1000",  
    "name": "jqsmith",  
    "email": "john.smith@example.org",  
    "enabled": true  
}
```

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

Name	Type	Description
id	String <i>(Required)</i>	The user ID.
name	String <i>(Required)</i>	The user name.
email	String <i>(Required)</i>	The user email.
enabled	Bool <i>(Required)</i>	Indicates whether the user is enabled (true) or disabled (false).

### Example 10.22. Update user: XML response

```
<?xml version="1.0" encoding="UTF-8"?>  
<user xmlns="http://docs.openstack.org/identity/api/v2.0"  
      enabled="true" email="john.smith@example.org" name="jqsmith"  
      id="u1000"/>
```

This operation does not return a response body.

## 10.4.3. 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)

### 10.4.3.1. Request

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

Name	Type	Description
{userId}	String	The ID of the user for which you want to perform the request.

This operation does not accept a request body.

## 10.4.4. Get user information by name

Method	URI	Description
GET	/v2.0/users/{name} {?name}	Gets detailed information about a specified user by user name.

**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)

### 10.4.4.1. Request

This operation does not accept a request body.

### 10.4.4.2. Response

#### Example 10.23. Get user information by name: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<user xmlns="http://docs.openstack.org/identity/api/v2.0"
      enabled="true" email="john.smith@example.org" name="jqsmith"
      id="u1000"/>
```

#### Example 10.24. Get user information by name: JSON response

```
{
    "user": {
        "id": "u1000",
        "name": "jqsmith",
        "email": "john.smith@example.org",
        "enabled": true
    }
}
```

## 10.4.5. Get user information by ID

Method	URI	Description
GET	/v2.0/users/{user_id}	Gets detailed information about a specified user by user ID.

**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)

### 10.4.5.1. 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.

### 10.4.5.2. Response

#### Example 10.25. Get user information by ID: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<user xmlns="http://docs.openstack.org/identity/api/v2.0"
      enabled="true" email="john.smith@example.org" name="jqsmith"
      id="u1000"/>
```

#### Example 10.26. Get user information by ID: JSON response

```
{
    "user": {
        "id": "u1000",
        "name": "jqsmith",
        "email": "john.smith@example.org",
        "enabled": true
    }
}
```

## 10.4.6. List user global roles

Method	URI	Description
GET	/v2.0/users/{user_id}/roles	Lists global roles for a specified 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)

### 10.4.6.1. Request

This table shows the URI parameters for the list user global roles request:

Name	Type	Description
{user_id}	String	The user ID.

This operation does not accept a request body.

### 10.4.6.2. Response

#### Example 10.27. List user global roles: JSON response

```
{
  "roles": [
    {
      "id": "123",
      "name": "compute:admin",
      "description": "Nova Administrator"
    }
  ],
  "roles_links": []
}
```

#### Example 10.28. List user global roles: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<roles xmlns="http://docs.openstack.org/identity/api/v2.0">
  <role id="123" name="Admin" description="All Access"/>
  <role id="234" name="Guest" description="Guest Access"/>
</roles>
```

## 10.5. Tenants

Method	URI	Description
GET	/v2.0/tenants{?marker,limit}	Lists all tenants.
GET	/v2.0/tenants{?marker,limit,name}	Gets detailed information about a specified tenant by name.
GET	/v2.0/tenants/{tenantId}	Gets detailed information about a specified tenant by ID.
GET	/v2.0/tenants/{tenantId}/users/{userId}/roles	Lists roles for a specified user on a specified tenant. Excludes global roles.



## 10.5.1. List tenants

Method	URI	Description
GET	/v2.0/tenants{?marker,limit}	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:** 200, 203

**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 list tenants 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.

### 10.5.1.2. Response

#### Example 10.29. 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": []
}
```

#### Example 10.30. Get tenants: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<tenants xmlns="http://docs.openstack.org/identity/api/v2.0">
    <tenant enabled="true" id="1234" name="ACME Corp">
```

```
<description>A description...</description>
</tenant>
<tenant enabled="true" id="3645" name="Iron Works">
    <description>A description...</description>
</tenant>
</tenants>
```

## 10.5.2. Get tenant information by name

Method	URI	Description
GET	/v2.0/tenants{?marker,limit,name}	Gets detailed information about a specified tenant by name.

**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)

### 10.5.2.1. 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
marker	String <i>(Optional)</i>	The ID of the last item in the previous list.
limit	Int <i>(Optional)</i>	The page size.
name	String <i>(Required)</i>	The name of the tenant.

This operation does not accept a request body.

### 10.5.2.2. Response

#### Example 10.31. Get tenant by name: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

#### Example 10.32. Get tenant by name: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<tenant xmlns="http://docs.openstack.org/identity/api/v2.0"
        enabled="true" id="1234" name="ACME Corp">
```

```
<description>A description...</description>
</tenant>
```

## 10.5.3. Get tenant information by ID

Method	URI	Description
GET	/v2.0/tenants/{tenantId}	Gets detailed information about a specified tenant by ID.

**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)

### 10.5.3.1. 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.

### 10.5.3.2. Response

#### Example 10.33. Get tenant by ID: JSON response

```
{
  "tenant": {
    "id": "1234",
    "name": "ACME corp",
    "description": "A description ...",
    "enabled": true
  }
}
```

#### Example 10.34. Get tenant by ID: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<tenant xmlns="http://docs.openstack.org/identity/api/v2.0"
        enabled="true" id="1234" name="ACME Corp">
  <description>A description...</description>
</tenant>
```

## 10.5.4. List roles for user

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/users/{userId}/roles	Lists roles for a specified user on a specified 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)

### 10.5.4.1. Request

This table shows the header parameters for the list roles for user request:

Name	Type	Description
X-Auth-Token	String <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 operation does not accept a request body.

### 10.5.4.2. Response

#### Example 10.35. List roles for user: JSON response

```
{
    "roles": [
        {
            "id": "123",
            "name": "compute:admin",
            "description": "Nova Administrator"
        }
    ],
    "roles_links": []
}
```

#### Example 10.36. List roles for user: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<roles xmlns="http://docs.openstack.org/identity/api/v2.0">
    <role id="123" name="Admin" description="All Access"/>
    <role id="234" name="Guest" description="Guest Access"/>
</roles>
```

# 11. Identity API v2.0 extensions (STABLE)

Query the Identity API to list available extensions with a **GET** request to `v2.0/extensions`.

Method	URI	Description
HP-IDM-serviceId extended parameter		
<b>GET</b>	<code>/v2.0/tokens/{tokenId}{?belongsTo, HP-IDM-serviceId}</code>	Validates that a token belongs to a specified tenant and services. Returns the permissions relevant to a particular client.
<b>HEAD</b>	<code>/v2.0/tokens/{tokenId}{?belongsTo, HP-IDM-serviceId}</code>	Validates that a token belongs to a specified tenant and services. For performance.
OS-KSADM admin extension		
<b>GET</b>	<code>/v2.0/users</code>	Lists users.
<b>POST</b>	<code>/v2.0/users</code>	Adds a user.
<b>PUT</b>	<code>/v2.0/users/{userId}</code>	Updates a user.
<b>DELETE</b>	<code>/v2.0/users/{userId}</code>	Deletes a user.
<b>GET</b>	<code>/v2.0/users/{userId}/roles{?serviceId,marker,limit}</code>	Lists global roles for a specified user.
<b>PUT</b>	<code>/v2.0/users/{userId}/roles/OS-KSADM/{roleId}</code>	Adds a specific global role to a user.
<b>DELETE</b>	<code>/v2.0/users/{userId}/roles/OS-KSADM/{roleId}</code>	Deletes a specific global role from a user.
<b>POST</b>	<code>/v2.0/tenants</code>	Creates a tenant.
<b>POST</b>	<code>/v2.0/tenants/{tenantId}</code>	Updates a tenant.
<b>DELETE</b>	<code>/v2.0/tenants/{tenantId}</code>	Deletes a tenant.
<b>GET</b>	<code>/v2.0/tenants/{tenantId}/users{?marker,limit}</code>	Lists all users for a tenant.
<b>PUT</b>	<code>/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}</code>	Adds a specified role to a user for a tenant.
<b>DELETE</b>	<code>/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}</code>	Deletes a specified role from a user on a tenant.
<b>GET</b>	<code>/v2.0/OS-KSADM/roles</code>	Gets a role by name.
<b>GET</b>	<code>/v2.0/OS-KSADM/roles/</code>	Lists roles.
<b>POST</b>	<code>/v2.0/OS-KSADM/roles</code>	Adds a role.
<b>GET</b>	<code>/v2.0/OS-KSADM/roles/{roleId}</code>	Gets information for a specified role.
<b>DELETE</b>	<code>/v2.0/OS-KSADM/roles/{roleId}</code>	Deletes a role.
<b>GET</b>	<code>/v2.0/OS-KSADM/services{?marker, limit}</code>	Lists services.
<b>POST</b>	<code>/v2.0/OS-KSADM/services{?marker, limit}</code>	Adds a service.
<b>GET</b>	<code>/v2.0/OS-KSADM/services/{?name}</code>	Gets a service by name.
<b>GET</b>	<code>/v2.0/OS-KSADM/services/{serviceId}</code>	Gets a service.
<b>DELETE</b>	<code>/v2.0/OS-KSADM/services/{serviceId}</code>	Deletes a service.
OS-KSCATALOG admin extension		
<b>GET</b>	<code>/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints</code>	Lists endpoints for a tenant.

Method	URI	Description
<b>POST</b>	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints	Adds endpoint to a tenant.
<b>GET</b>	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints/{endpointId}	Gets endpoint for a tenant.
<b>DELETE</b>	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints/{endpointId}	Deletes an endpoint from a tenant.
<b>GET</b>	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Lists endpoint templates.
<b>POST</b>	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Adds endpoint template.
<b>GET</b>	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Gets endpoint templates.
<b>PUT</b>	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Updates endpoint template.
<b>DELETE</b>	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Deletes an endpoint template.
OS-KSEC2 admin extension		
<b>GET</b>	/v2.0/users/{userId}/OS-KSADM/credentials{?marker,limit}	Lists credentials.
<b>POST</b>	/v2.0/users/{userId}/OS-KSADM/credentials{?marker,limit}	Adds a credential to a user.
<b>GET</b>	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Gets user credentials.
<b>POST</b>	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Updates credentials for a specified user.
<b>DELETE</b>	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Deletes user credentials.
<b>GET</b>	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials/{type}{?type}	Lists credentials by type.
OS-KSS3 admin extension		
<b>GET</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials{?marker,limit}	Lists credentials.
<b>POST</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials{?marker,limit}	Adds a credential to a user.
<b>GET</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials/s3credentials	Gets user credentials.
<b>POST</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials/s3credentials	Updates credentials.
<b>DELETE</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials/s3credentials	Deletes user credentials.
<b>GET</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials/s3credentials/{type}{?type}	Lists credentials by type.
OS-KVALIDATE admin extension		
<b>GET</b>	/v2.0/OS-KVALIDATE/token/validate{?belongsTo,HP-IDM-serviceId}	Checks that a token is valid and that it belongs to a specified tenant and service IDs. Returns the permissions for a particular client.
<b>HEAD</b>	/v2.0/OS-KVALIDATE/token/validate{?belongsTo,HP-IDM-serviceId}	Checks that a token is valid and that it belongs to a specified tenant and service IDs, for performance.
<b>GET</b>	/v2.0/OS-KVALIDATE/token/endpoints{?HP-IDM-serviceId}	Lists endpoints associated with a specific token.
RAX-GRPADM admin extension		

Method	URI	Description
<b>GET</b>	/v2.0/RAX-GRPADM/groups{?marker, limit, name}	Lists groups.
<b>POST</b>	/v2.0/RAX-GRPADM/groups{?marker, limit, name}	Adds a group.
<b>GET</b>	/v2.0/RAX-GRPADM/groups/{groupId}	Gets information for a group by ID.
<b>PUT</b>	/v2.0/RAX-GRPADM/groups/{groupId}	Updates a group.
<b>DELETE</b>	/v2.0/RAX-GRPADM/groups/{groupId}	Deletes a group.
<b>GET</b>	/v2.0/RAX-GRPADM/groups/{groupId}/users{?marker, limit}	Lists users for a group.
<b>PUT</b>	/v2.0/RAX-GRPADM/groups/{groupId}/users/{userId}	Adds a user to a group.
<b>DELETE</b>	/v2.0/RAX-GRPADM/groups/{groupId}/users/{userId}	Removes a user from a group.
RAX-KSGRP admin extension		
<b>GET</b>	/v2.0/users/{userId}/RAX-KSGRP	Lists groups for a user.
RAX-KSKEY admin extension		
<b>POST</b>	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials	Adds a credential to a user.
<b>GET</b>	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials{?marker, limit}	Lists credentials.
<b>POST</b>	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials/RAX-KSKEY:apiKeyCredentials	Updates credentials.
<b>DELETE</b>	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials/RAX-KSKEY:apiKeyCredentials	Deletes user credentials.
<b>GET</b>	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials/RAX-KSKEY:apiKeyCredentials	Gets user credentials.
RAX-KSQA admin extension		
<b>GET</b>	/v2.0/users/{userId}/RAX-KSQA/se-cretqa	Gets a secret question and answer for a specified user.
<b>PUT</b>	/v2.0/users/{userId}/RAX-KSQA/se-cretqa	Updates a secret question and answer for a specified user.

## 11.1. HP-IDM-serviceId extended parameter

Method	URI	Description
<b>GET</b>	/v2.0/tokens/{tokenId}{?belongsTo, HP-IDM-serviceId}	Validates that a token belongs to a specified tenant and services. Returns the permissions relevant to a particular client.
<b>HEAD</b>	/v2.0/tokens/{tokenId}{?belongsTo, HP-IDM-serviceId}	Validates that a token belongs to a specified tenant and services. For performance.

## 11.1.1. Validate token

Method	URI	Description
GET	/v2.0/tokens/{tokenId}{?belongsTo, HP-IDM-serviceId}	Validates that a token belongs to a specified tenant and services. Returns the permissions relevant to a particular client.

**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)

### 11.1.1.1. 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.

This table shows the URI parameters for the validate token request:

Name	Type	Description
{tokenId}	String	The token ID.

This operation does not accept a request body.

### 11.1.1.2. Response

#### Example 11.1. 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": [
]
}
}
```

### Example 11.2. Validate token: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<access xmlns="http://docs.openstack.org/identity/api/v2.0">
    <token id="ab48a9efdfedb23ty3494"
        expires="2010-11-01T03:32:15-05:00">
        <tenant id="456" name="My Project"/>
    </token>
    <user id="123" username="jqsmith">
        <roles
            xmlns="http://docs.openstack.org/identity/api/v2.0">
            <role id="123" name="Admin" tenantId="one"/>
            <role id="234" name="object-store:admin"
                tenantId="1"/>
        </roles>
    </user>
</access>
```

## 11.1.2. Validate token for performance

Method	URI	Description
<b>HEAD</b>	/v2.0/tokens/{tokenId}{?belongsTo, HP-IDM-serviceId}	Validates that a token belongs to a specified tenant and services. For performance.

**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)

### 11.1.2.1. Request

This table shows the header parameters for the validate token for performance request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

This table shows the URI parameters for the validate token for performance request:

Name	Type	Description
{tokenId}	String	The token ID.

This operation does not accept a request body.

## 11.2. OS-KSADM admin extension

Method	URI	Description
<b>GET</b>	/v2.0/users	Lists users.
<b>POST</b>	/v2.0/users	Adds a user.
<b>PUT</b>	/v2.0/users/{userId}	Updates a user.
<b>DELETE</b>	/v2.0/users/{userId}	Deletes a user.
<b>GET</b>	/v2.0/users/{userId}/roles{?serviceId,marker,limit}	Lists global roles for a specified user.
<b>PUT</b>	/v2.0/users/{userId}/roles/OS-KSADM/{roleId}	Adds a specific global role to a user.
<b>DELETE</b>	/v2.0/users/{userId}/roles/OS-KSADM/{roleId}	Deletes a specific global role from a user.
<b>POST</b>	/v2.0/tenants	Creates a tenant.
<b>POST</b>	/v2.0/tenants/{tenantId}	Updates a tenant.
<b>DELETE</b>	/v2.0/tenants/{tenantId}	Deletes a tenant.
<b>GET</b>	/v2.0/tenants/{tenantId}/users{?marker,limit}	Lists all users for a tenant.
<b>PUT</b>	/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}	Adds a specified role to a user for a tenant.
<b>DELETE</b>	/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}	Deletes a specified role from a user on a tenant.

Method	URI	Description
<b>GET</b>	/v2.0/OS-KSADM/roles	Gets a role by name.
<b>GET</b>	/v2.0/OS-KSADM/roles/	Lists roles.
<b>POST</b>	/v2.0/OS-KSADM/roles	Adds a role.
<b>GET</b>	/v2.0/OS-KSADM/roles/{roleId}	Gets information for a specified role.
<b>DELETE</b>	/v2.0/OS-KSADM/roles/{roleId}	Deletes a role.
<b>GET</b>	/v2.0/OS-KSADM/services{?marker, limit}	Lists services.
<b>POST</b>	/v2.0/OS-KSADM/services{?marker, limit}	Adds a service.
<b>GET</b>	/v2.0/OS-KSADM/services/{?name}	Gets a service by name.
<b>GET</b>	/v2.0/OS-KSADM/services/{serviceId}	Gets a service.
<b>DELETE</b>	/v2.0/OS-KSADM/services/{serviceId}	Deletes a service.

## 11.2.1. List Users

Method	URI	Description
GET	/v2.0/users	Lists users.

**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)

### 11.2.1.1. 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 operation does not accept a request body.

### 11.2.1.2. Response

#### Example 11.3. List Users: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<users xmlns="http://docs.openstack.org/identity/api/v2.0">
    <user xmlns="http://docs.openstack.org/identity/api/v2.0"
        enabled="true" email="john.smith@example.org"
        name="jqsmith" id="u1000"/>
    <user xmlns="http://docs.openstack.org/identity/api/v2.0"
        enabled="true" email="john.smith@example.org"
        name="jqsmith" id="u1001"/>
</users>
```

#### Example 11.4. List Users: JSON response

```
{
    "users": [
        {
            "id": "u1000",
            "name": "jqsmith",
            "email": "john.smith@example.org",
            "enabled": true
        },
        {
            "id": "u1001",
            "name": "jqsmith",
            "email": "john.smith@example.org",
            "enabled": true
        }
    ],
    "users_links": []
}
```



## 11.2.2. Add User

Method	URI	Description
POST	/v2.0/users	Adds 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)

### 11.2.2.1. Request

This table shows the header parameters for the add user request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the body parameters for the add user request:

Name	Type	Description
name	String <i>(Optional)</i>	The user name.

#### Example 11.5. Add User: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<user xmlns="http://docs.openstack.org/identity/api/v2.0"
      xmlns:OS-KSADM="http://docs.openstack.org/identity/api/ext/OS-KSADM/v1.
0"
      enabled="true" email="john.smith@example.org"
      name="jqsmith"
      OS-KSADM:password="secrete"/>
```

#### Example 11.6. Add User: JSON request

```
{
  "user": {
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true,
    "OS-KSADM:password": "secrete"
  }
}
```

### 11.2.2.2. Response

#### Example 11.7. Add User: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<user xmlns="http://docs.openstack.org/identity/api/v2.0"
```

```
    enabled="true" email="john.smith@example.org"
    name="jqsmith" id="u1000"/>
```

### Example 11.8. Add User: JSON response

```
{
  "user": {
    "id": "u1000",
    "name": "jqsmith",
    "email": "john.smith@example.org",
    "enabled": true
  }
}
```

## 11.2.3. 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), badMediaType (415), itemNotFound (404)

### 11.2.3.1. 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 you want to perform the request.

#### Example 11.9. Update User: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<user xmlns="http://docs.openstack.org/identity/api/v2.0"
      enabled="true" email="john.smith@example.org"
      name="jqsmith" id="u1000"/>
```

#### Example 11.10. Update User: JSON request

```
{
    "user": {
        "id": "u1000",
        "name": "jqsmith",
        "email": "john.smith@example.org",
        "enabled": true
    }
}
```

### 11.2.3.2. Response

#### Example 11.11. Update User: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<user xmlns="http://docs.openstack.org/identity/api/v2.0"
      enabled="true" email="john.smith@example.org"
      name="jqsmith" id="u1000"/>
```

#### Example 11.12. Update User: JSON response

```
{
```

```
    "user": {  
        "id": "u1000",  
        "name": "jqsmith",  
        "email": "john.smith@example.org",  
        "enabled": true  
    }  
}
```

## 11.2.4. 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)

### 11.2.4.1. 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 you want to perform the request.

This operation does not accept a request body.

## 11.2.5. List global roles for user

Method	URI	Description
GET	/v2.0/users/{userId}/roles{?serviceId,marker,limit}	Lists global roles for a specified user.

**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)

### 11.2.5.1. 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 you want to perform the request.

This operation does not accept a request body.

### 11.2.5.2. Response

#### Example 11.13. List global roles for user: JSON response

```
{
  "roles": [
    {
      "id": "8341d3603a1d4d5985bff09f10704d4d",
      "name": "service"
    },
    {
      "id": "2e66d57df76946fdbbe034bc4da6fdec0",
      "name": "admin"
    }
  ]
}
```

#### Example 11.14. List global roles for user: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<roles xmlns="http://docs.openstack.org/identity/api/v2.0">
  <role id="8341d3603a1d4d5985bff09f10704d4d" name="service"/>
  <role id="2e66d57df76946fdbbe034bc4da6fdec0" name="admin"/>
</roles>
```

## 11.2.6. Add global role to user

Method	URI	Description
PUT	/v2.0/users/{userId}/roles/OS-KSADM/{roleId}	Adds a specific 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)

### 11.2.6.1. Request

This table shows the header parameters for the add 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 add global role to user request:

Name	Type	Description
{userId}	String	The ID of the user for which you want to perform the request.
{roleId}	Int	The ID of the role that you want to add or delete.

This operation does not accept a request body.

## 11.2.7. Delete Global Role from User

Method	URI	Description
DELETE	/v2.0/users/{userId}/roles/OS-KSADM/{roleId}	Deletes a specific 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)

### 11.2.7.1. 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 you want to perform the request.
{roleId}	Int	The ID of the role that you want to add or delete.

This operation does not accept a request body.

## 11.2.8. Add 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)

### 11.2.8.1. Request

This table shows the header parameters for the add tenant request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 11.15. Add Tenant: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<tenant xmlns="http://docs.openstack.org/identity/api/v2.0"
         enabled="true" name="ACME Corp">
    <description>A description...</description>
</tenant>
```

#### Example 11.16. Add Tenant: JSON request

```
{
    "tenant": {
        "name": "ACME corp",
        "description": "A description ...",
        "enabled": true
    }
}
```

### 11.2.8.2. Response

#### Example 11.17. Add Tenant: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<tenant xmlns="http://docs.openstack.org/identity/api/v2.0"
         enabled="true" id="1234" name="ACME Corp">
    <description>A description...</description>
</tenant>
```

#### Example 11.18. Add Tenant: JSON response

```
{
    "tenant": {
        "id": "1234",
        "name": "ACME corp",
```

```
        "description": "A description ...",
        "enabled": true
    }
```

## 11.2.9. 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)

### 11.2.9.1. 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 11.19. Update Tenant: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<tenant xmlns="http://docs.openstack.org/identity/api/v2.0"
         enabled="true" id="1234" name="ACME Corp">
    <description>A description...</description>
</tenant>
```

#### Example 11.20. Update Tenant: JSON request

```
{
    "tenant": {
        "id": "1234",
        "name": "ACME corp",
        "description": "A description ...",
        "enabled": true
    }
}
```

### 11.2.9.2. Response

#### Example 11.21. Update Tenant: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<tenant xmlns="http://docs.openstack.org/identity/api/v2.0"
         enabled="true" id="1234" name="ACME Corp">
    <description>A description...</description>
</tenant>
```

**Example 11.22. Update Tenant: JSON response**

```
{  
    "tenant": {  
        "id": "1234",  
        "name": "ACME corp",  
        "description": "A description ...",  
        "enabled": true  
    }  
}
```

## 11.2.10. 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)

### 11.2.10.1. 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.

## 11.2.11. List Users for a Tenant

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/users{?marker,limit}	Lists all users 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)

### 11.2.11.1. Request

This table shows the header parameters for the list users for 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 for a tenant request:

Name	Type	Description
{tenantId}	String	The tenant ID.

This operation does not accept a request body.

### 11.2.11.2. Response

#### Example 11.23. List Users for a Tenant: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<users xmlns="http://docs.openstack.org/identity/api/v2.0">
    <user xmlns="http://docs.openstack.org/identity/api/v2.0"
          enabled="true" email="john.smith@example.org"
          name="jqsmith" id="u1000"/>
    <user xmlns="http://docs.openstack.org/identity/api/v2.0"
          enabled="true" email="john.smith@example.org"
          name="jqsmith" id="u1001"/>
</users>
```

#### Example 11.24. List Users for a Tenant: JSON response

```
{
    "users": [
        {
            "id": "u1000",
            "name": "jqsmith",
            "email": "john.smith@example.org",
            "enabled": true
        },
        {
            "id": "u1001",
```

```
        "name": "jqsmith",
        "email": "john.smith@example.org",
        "enabled": true
    }
],
"users_links": []
}
```

## 11.2.12. Add Roles to User on Tenant

Method	URI	Description
PUT	/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}	Adds a specified 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)

### 11.2.12.1. Request

This table shows the header parameters for the add 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 add roles to user on tenant request:

Name	Type	Description
{tenantId}	String	The tenant ID.
{userId}	String	The user ID.
{roleId}	String	The role ID.

This operation does not accept a request body.

## 11.2.13. Delete Roles from User on Tenant

Method	URI	Description
DELETE	/v2.0/tenants/{tenantId}/users/{userId}/roles/OS-KSADM/{roleId}	Deletes a specified role from a user on 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)

### 11.2.13.1. Request

This table shows the header parameters for the delete roles 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 delete roles from user on tenant request:

Name	Type	Description
{tenantId}	String	The tenant ID.
{userId}	String	The user ID.
{roleId}	String	The role ID.

This operation does not accept a request body.

## 11.2.14. Get role information by name

Method	URI	Description
GET	/v2.0/OS-KSADM/roles	Gets a role by name.

**Normal response codes:** 200, 203

**Error response codes:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

### 11.2.14.1. Request

This table shows the header parameters for the get role information by name 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.

### 11.2.14.2. Response

#### Example 11.25. Get role information by name: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<role xmlns="http://docs.openstack.org/identity/api/v2.0"
      id="123" name="Admin" description="All Access" />
```

#### Example 11.26. Get role information by name: JSON response

```
{
    "role": {
        "id": "123",
        "name": "Guest",
        "description": "Guest Access"
    }
}
```

## 11.2.15. List Roles

Method	URI	Description
GET	/v2.0/OS-KSADM/roles/	Lists 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)

### 11.2.15.1. 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 operation does not accept a request body.

### 11.2.15.2. Response

#### Example 11.27. List Roles: XML response

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

<roles xmlns="http://docs.openstack.org/identity/api/v2.0">
    <role id="123" name="Admin" description="All Access" />
    <role id="234" name="Guest" description="Guest Access" />
</roles>
```

#### Example 11.28. List Roles: JSON response

```
{
    "roles": [
        {
            "id": "123",
            "name": "compute:admin",
            "description": "Nova Administrator"
        }
    ],
    "roles_links": []
}
```

## 11.2.16. Add Role

Method	URI	Description
POST	/v2.0/OS-KSADM/roles	Adds 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)

### 11.2.16.1. Request

This table shows the header parameters for the add role request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 11.29. Add Role: XML request

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

<role xmlns="http://docs.openstack.org/identity/api/v2.0"
      id="123" name="Admin" description="All Access" />
```

#### Example 11.30. Add Role: JSON request

```
{
    "role": {
        "id": "123",
        "name": "Guest",
        "description": "Guest Access"
    }
}
```

### 11.2.16.2. Response

#### Example 11.31. Add Role: XML response

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

<role xmlns="http://docs.openstack.org/identity/api/v2.0"
      id="123" name="Admin" description="All Access" />
```

#### Example 11.32. Add Role: JSON response

```
{
    "role": {
        "id": "123",
        "name": "Guest",
        "description": "Guest Access"
    }
}
```

}

## 11.2.17. Get role information

Method	URI	Description
GET	/v2.0/OS-KSADM/roles/{roleId}	Gets information for a specified role.

**Normal response codes:** 200, 203

**Error response codes:** identityFault (400, 500, ...), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), serviceUnavailable (503), badMedia (415), itemNotFound (404)

### 11.2.17.1. Request

This table shows the header parameters for the get role information 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 role information request:

Name	Type	Description
{roleId}	String	The role ID.

This operation does not accept a request body.

### 11.2.17.2. Response

#### Example 11.33. Get role information: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<role xmlns="http://docs.openstack.org/identity/api/v2.0"
      id="123" name="Admin" description="All Access" />
```

#### Example 11.34. Get role information: JSON response

```
{
    "role": {
        "id": "123",
        "name": "Guest",
        "description": "Guest Access"
    }
}
```

## 11.2.18. Delete Role

Method	URI	Description
DELETE	/v2.0/OS-KSADM/roles/{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)

### 11.2.18.1. 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.

## 11.2.19. List Services

Method	URI	Description
GET	/v2.0/OS-KSADM/services{?marker, limit}	Lists services.

**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)

### 11.2.19.1. 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 operation does not accept a request body.

### 11.2.19.2. Response

#### Example 11.35. List Services: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<services
  xmlns="http://docs.openstack.org/identity/api/ext/OS-KSADM/v1.0">
  <service id="123" name="nova" type="compute"
    description="OpenStack Compute Service"/>
  <service id="234" name="glance" type="image"
    description="OpenStack Image Service"/>
</services>
```

#### Example 11.36. List Services: JSON response

```
{
  "OS-KSADM:services": [
    {
      "id": "123",
      "name": "nova",
      "type": "compute",
      "description": "OpenStack Compute Service"
    },
    {
      "id": "234",
      "name": "glance",
      "type": "image",
      "description": "OpenStack Image Service"
    }
  ],
  "OS-KSADM:services_links": []
}
```



## 11.2.20. Add Service

Method	URI	Description
POST	/v2.0/OS-KSADM/services{?marker, limit}	Adds 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)

### 11.2.20.1. Request

This table shows the header parameters for the add service request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 11.37. Add Service: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<service
  xmlns="http://docs.openstack.org/identity/api/ext/OS-KSADM/v1.0"
  id="123" name="nova" type="compute"
  description="OpenStack Compute Service"/>
```

#### Example 11.38. Add Service: JSON request

```
{
  "OS-KSADM:service": {
    "id": "123",
    "name": "nova",
    "type": "compute",
    "description": "OpenStack Compute Service"
  }
}
```

### 11.2.20.2. Response

#### Example 11.39. Add Service: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<service
  xmlns="http://docs.openstack.org/identity/api/ext/OS-KSADM/v1.0"
  id="123" name="nova" type="compute"
  description="OpenStack Compute Service"/>
```

#### Example 11.40. Add Service: JSON response

```
{
  "OS-KSADM:service": {
```

```
        "id": "123",
        "name": "nova",
        "type": "compute",
        "description": "OpenStack Compute Service"
    }
}
```

## 11.2.21. Get Service by Name

Method	URI	Description
GET	/v2.0/OS-KSADM/services/{?name}	Gets a service by name.

**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)

### 11.2.21.1. Request

This table shows the header parameters for the get service by name 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.

### 11.2.21.2. Response

#### Example 11.41. Get Service by Name: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<service
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSADM/v1.0"
    id="123" name="nova" type="compute"
    description="OpenStack Compute Service"/>
```

#### Example 11.42. Get Service by Name: JSON response

```
{
    "OS-KSADM:service": {
        "id": "123",
        "name": "nova",
        "type": "compute",
        "description": "OpenStack Compute Service"
    }
}
```

## 11.2.22. Get Service

Method	URI	Description
GET	/v2.0/OS-KSADM/services/{serviceId}	Gets a service.

**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)

### 11.2.22.1. Request

This table shows the header parameters for the get 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 get service request:

Name	Type	Description
{serviceId}	String	The service ID.

This operation does not accept a request body.

### 11.2.22.2. Response

#### Example 11.43. Get Service: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<service
  xmlns="http://docs.openstack.org/identity/api/ext/OS-KSADM/v1.0"
  id="123" name="nova" type="compute"
  description="OpenStack Compute Service"/>
```

#### Example 11.44. Get Service: JSON response

```
{
  "OS-KSADM:service": {
    "id": "123",
    "name": "nova",
    "type": "compute",
    "description": "OpenStack Compute Service"
  }
}
```

## 11.2.23. Delete Service

Method	URI	Description
<b>DELETE</b>	/v2.0/OS-KSADM/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)

### 11.2.23.1. 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.

## 11.3. OS-KSCATALOG admin extension

Method	URI	Description
<b>GET</b>	/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints	Lists endpoints for a tenant.
<b>POST</b>	/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints	Adds endpoint to a tenant.
<b>GET</b>	/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints/{endpointId}	Gets endpoint for a tenant.
<b>DELETE</b>	/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints/{endpointId}	Deletes an endpoint from a tenant.
<b>GET</b>	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Lists endpoint templates.
<b>POST</b>	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Adds endpoint template.
<b>GET</b>	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Gets endpoint templates.
<b>PUT</b>	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Updates endpoint template.
<b>DELETE</b>	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Deletes an endpoint template.

## 11.3.1. List endpoints

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints	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)

### 11.3.1.1. 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 operation does not accept a request body.

### 11.3.1.2. Response

#### Example 11.45. List endpoints: XML response

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

<endpoints
    xmlns="http://docs.openstack.org/identity/api/v2.0">
    <endpoint
        id="1"
        tenantId="1"
        type="compute"
        name="Compute"
        region="North"
        publicURL="https://compute.north.public.com/v1"
        internalURL="https://compute.north.internal.com/v1"
        adminURL="https://compute.north.internal.com/v1">
        <version
            id="1"
            info="https://compute.north.public.com/v1/"
            list="https://compute.north.public.com/">
        />
    </endpoint>
    <endpoint
        id="2"
        tenantId="2"
        type="compute"
        name="Compute"
```

```
region="South"
publicURL="https://compute.north.public.com/v1"
internalURL="https://compute.north.internal.com/v1"
adminURL="https://compute.north.internal.com/v1">
<version
    id="1"
    info="https://compute.north.public.com/v1/"
    list="https://compute.north.public.com/">
</version>
</endpoint>
<endpoint
    id="3"
    tenantId="1"
    type="compute"
    name="Compute"
    region="East"
    publicURL="https://compute.north.public.com/v1"
    internalURL="https://compute.north.internal.com/v1"
    adminURL="https://compute.north.internal.com/v1">
</version>
<endpoint
    id="4"
    tenantId="1"
    type="compute"
    name="Compute"
    region="West"
    publicURL="https://compute.north.public.com/v1"
    internalURL="https://compute.north.internal.com/v1"
    adminURL="https://compute.north.internal.com/v1">
<version
    id="1"
    info="https://compute.north.public.com/v1/"
    list="https://compute.north.public.com/">
</version>
</endpoint>
<endpoint
    id="5"
    tenantId="1"
    type="compute"
    name="Compute"
    region="Global"
    publicURL="https://compute.north.public.com/v1"
    internalURL="https://compute.north.internal.com/v1"
    adminURL="https://compute.north.internal.com/v1">
<version
    id="1"
    info="https://compute.north.public.com/v1/"
    list="https://compute.north.public.com/">
</version>
</endpoint>
</endpoints>
```

#### Example 11.46. 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": []
}
```

## 11.3.2. Add endpoint

Method	URI	Description
POST	/v2.0/tenants/{tenantId}/OS-KSCATALOG/endpoints	Adds 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)

### 11.3.2.1. Request

This table shows the header parameters for the add 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 add endpoint request:

Name	Type	Description
{tenantId}	String	The tenant ID.

This table shows the body parameters for the add endpoint request:

Name	Type	Description
endpoint	Endpoint Template WithOnly Id <i>(Optional)</i>	

#### Example 11.47. Add endpoint: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<endpointTemplate
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSCATALOG/v1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="EndpointTemplateWithOnlyId"
    id="1"/>
```

#### Example 11.48. Add endpoint: JSON request

```
{
    "OS-KSCATALOG:endpointTemplate": {
        "id": 1
    }
}
```

### 11.3.2.2. Response

#### Example 11.49. Add endpoint: XML response

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

<endpoint
    id="1"
    tenantId="1"
    type="compute"
    name="Compute"
    region="North"
    publicURL="https://compute.north.public.com/v1"
    internalURL="https://compute.north.internal.com/v1"
    adminURL="https://compute.north.internal.com/v1"
    xmlns="http://docs.openstack.org/identity/api/v2.0">
    <version
        id="1"
        info="https://compute.north.public.com/v1/"
        list="https://compute.north.public.com/"
    />
</endpoint>
```

#### Example 11.50. Add 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/"
    }
}
```

### 11.3.3. Get endpoint

Method	URI	Description
GET	/v2.0/tenants/{tenantId}/OS-KSCAT-ALOG/endpoints/{endpointId}	Gets endpoint 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)

#### 11.3.3.1. 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.

#### 11.3.3.2. Response

##### Example 11.51. Get endpoint: XML response

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

<endpoint
    id="1"
    tenantId="1"
    type="compute"
    name="Compute"
    region="North"
    publicURL="https://compute.north.public.com/v1"
    internalURL="https://compute.north.internal.com/v1"
    adminURL="https://compute.north.internal.com/v1"
    xmlns="http://docs.openstack.org/identity/api/v2.0">
    <version
        id="1"
        info="https://compute.north.public.com/v1/"
        list="https://compute.north.public.com/" />
</endpoint>
```

##### Example 11.52. 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/"
    }
}
```

## 11.3.4. 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)

### 11.3.4.1. 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.

## 11.3.5. List endpoint templates

Method	URI	Description
GET	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Lists endpoint templates.

**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)

### 11.3.5.1. 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 operation does not accept a request body.

### 11.3.5.2. Response

#### Example 11.53. List endpoint templates: XML response

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

<endpointTemplates xmlns="http://docs.openstack.org/identity/api/ext/OS-KSCATALOG/v1.0">
    <endpointTemplate
        id="1"
        region="North"
        global="true"
        type="compute"
        name="Compute"
        publicURL="https://compute.north.public.com/v1"
        internalURL="https://compute.north.internal.com/v1"
        enabled="true">
        <version
            id="1"
            list="https://compute.north.public.com/"
            info="https://compute.north.public.com/v1"/>
    </endpointTemplate>
    <endpointTemplate
        id="2"
        region="south"
        type="compute"
        name="Compute"
        publicURL="https://service2.public.com/v1"
        internalURL="https://service2.internal.public.com/v1"
        enabled="false">
        <version
            id="1"
            list="https://service2.public.com/"
            info="https://service2.internal.public.com/v1"/>
    </endpointTemplate>
</endpointTemplates>
```

```
list="https://service1.public.com/"
info="https://service1.public.com/v1"/>
</endpointTemplate>
<endpointTemplate
id="3"
region="DFW"
global="true"
type="ext1:service1"
name="Compute"
publicURL="https://service1.public.com/v1"
enabled="true">
<version
id="1"
list="https://service1.public.com/"
info="https://service1.public.com/v1"/>
</endpointTemplate>
<endpointTemplate
id="4"
region="ORD"
type="compute"
name="Compute"
publicURL="https://service2.public.com/v1"
enabled="true">
<version
id="1"
list="https://service1.public.com/"
info="https://service1.public.com/v1"/>
</endpointTemplate>
<endpointTemplate
id="5"
global="true"
type="compute"
name="Compute"
publicURL="https://service3.public.com/v1">
<version
id="1"
list="https://service1.public.com/"
info="https://service1.public.com/v1"/>
</endpointTemplate>
</endpointTemplates>
```

#### Example 11.54. 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": []
}
```

## 11.3.6. Add endpoint template

Method	URI	Description
POST	/v2.0/OS-KSCATALOG/endpointTemplates{?serviceId}	Adds 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)

### 11.3.6.1. Request

This table shows the header parameters for the add endpoint template request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

#### Example 11.55. Add endpoint template: XML request

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

<endpointTemplate
  xmlns="http://docs.openstack.org/identity/api/ext/OS-KSCATALOG/v1.0"
  id="1"
  region="North"
  global="true"
  type="compute"
  name="Compute"
  publicURL="https://service-public.com/v1"
  internalURL="https://service-internal.com/v1"
  enabled="true">
  <version
    id="1"
    info="https://compute.north.public.com/v1/"
    list="https://compute.north.public.com/"
  />
</endpointTemplate>
```

#### Example 11.56. Add endpoint template: JSON request

```
{
  "OS-KSCATALOG:endpointTemplate": {
    "id": 1,
    "region": "North",
    "global": true,
    "type": "compute",
    "publicURL": "https://compute.north.public.com/v1",
    "internalURL": "https://compute.north.internal.com/v1",
    "versionId": "1",
    "versionInfo": "https://compute.north.public.com/v1/",
    "versionList": "https://compute.north.public.com/",
    "enabled": true
  }
}
```

```
    }  
}
```

### 11.3.6.2. Response

#### Example 11.57. Add endpoint template: XML response

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<endpointTemplate  
  xmlns="http://docs.openstack.org/identity/api/ext/OS-KSCATALOG/v1.0"  
  id="1"  
  region="North"  
  global="true"  
  type="compute"  
  name="Compute"  
  publicURL="https://service-public.com/v1"  
  internalURL="https://service-internal.com/v1"  
  enabled="true">  
  <version  
    id="1"  
    info="https://compute.north.public.com/v1/"  
    list="https://compute.north.public.com/"  
  />  
</endpointTemplate>
```

#### Example 11.58. Add 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  
  }  
}
```

## 11.3.7. Get endpoint template

Method	URI	Description
GET	/v2.0/OS-KSCATALOG/endpointTemplates/{endpointTemplateId}	Gets endpoint templates.

**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)

### 11.3.7.1. 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.

### 11.3.7.2. Response

#### Example 11.59. Get endpoint template: XML response

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

<endpointTemplate
  xmlns="http://docs.openstack.org/identity/api/ext/OS-KSCATALOG/v1.0"
  id="1"
  region="North"
  global="true"
  type="compute"
  name="Compute"
  publicURL="https://service-public.com/v1"
  internalURL="https://service-internal.com/v1"
  enabled="true">
  <version
    id="1"
    info="https://compute.north.public.com/v1/"
    list="https://compute.north.public.com/"
  />
</endpointTemplate>
```

#### Example 11.60. 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  
}  
}
```

## 11.3.8. 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)

### 11.3.8.1. 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 11.61. Update endpoint template: XML request

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

<endpointTemplate
  xmlns="http://docs.openstack.org/identity/api/ext/OS-KSCATALOG/v1.0"
  id="1"
  region="North"
  global="true"
  type="compute"
  name="Compute"
  publicURL="https://service-public.com/v1"
  internalURL="https://service-internal.com/v1"
  enabled="true">
  <version
    id="1"
    info="https://compute.north.public.com/v1/"
    list="https://compute.north.public.com/"
  />
</endpointTemplate>
```

#### Example 11.62. Update endpoint template: JSON request

```
{
  "OS-KSCATALOG:endpointTemplate": {
    "id": 1,
    "region": "North",
    "global": true,
    "type": "compute",
```

```
        "publicURL": "https://compute.north.public.com/v1",
        "internalURL": "https://compute.north.internal.com/v1",
        "versionId": "1",
        "versionInfo": "https://compute.north.public.com/v1/",
        "versionList": "https://compute.north.public.com/",
        "enabled": true
    }
}
```

### 11.3.8.2. Response

#### Example 11.63. Update endpoint template: XML response

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

<endpointTemplate
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSCATALOG/v1.0"
    id="1"
    region="North"
    global="true"
    type="compute"
    name="Compute"
    publicURL="https://service-public.com/v1"
    internalURL="https://service-internal.com/v1"
    enabled="true">
    <version
        id="1"
        info="https://compute.north.public.com/v1/"
        list="https://compute.north.public.com/"
    />
</endpointTemplate>
```

#### Example 11.64. 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
    }
}
```

## 11.3.9. Delete endpoint template.

Method	URI	Description
<b>DELETE</b>	/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)

### 11.3.9.1. 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.

## 11.4. OS-KSEC2 admin extension

Method	URI	Description
<b>GET</b>	/v2.0/users/{userId}/OS-KSADM/credentials{?marker,limit}	Lists credentials.
<b>POST</b>	/v2.0/users/{userId}/OS-KSADM/credentials{?marker,limit}	Adds a credential to a user.
<b>GET</b>	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Gets user credentials.
<b>POST</b>	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Updates credentials for a specified user.
<b>DELETE</b>	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Deletes user credentials.
<b>GET</b>	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials/{type}{?type}	Lists credentials by type.

## 11.4.1. List Credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSADM/credentials{?marker,limit}	Lists credentials.

**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)

### 11.4.1.1. 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 operation does not accept a request body.

### 11.4.1.2. Response

#### Example 11.65. List Credentials: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<credentials xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://docs.openstack.org/identity/api/v2.0">
  <passwordCredentials username="test_user" password="test"/>
  <ec2Credentials xmlns="http://docs.openstack.org/identity/api/ext/OS-
KSEC2/v1.0"
    username="testuser" key="aaaaaa" signature="bbbbbb"/>
</credentials>
```

#### Example 11.66. List Credentials: JSON response

```
{
  "credentials": [
    {
      "passwordCredentials": {
        "username": "test_user",
        "password": "mypass"
      }
    },
    {
      "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "key": "aaaaaa",
        "signature": "bbbbbb"
      }
    }
  ]
}
```

```
        "secret": "aaaaa",
        "signature": "bbb"
    }
},
"credentials_links": []
}
```

## 11.4.2. Add User Credentials

Method	URI	Description
POST	/v2.0/users/{userId}/OS-KSADM/credentials{?marker,limit}	Adds 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)

### 11.4.2.1. Request

This table shows the header parameters for the add 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 add user credentials request:

Name	Type	Description
{userId}	String	The user ID.

#### Example 11.67. Add User Credentials: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<ec2Credentials
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSEC2/v1.0"
    username="testuser"
    key="aaaaaa"
    signature="bbbbbb" />
```

#### Example 11.68. Add User Credentials: JSON request

```
{
    "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "secret": "aaaaaa",
        "signature": "bbb"
    }
}
```

### 11.4.2.2. Response

#### Example 11.69. Add User Credentials: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ec2Credentials
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSEC2/v1.0"
    username="testuser"
    key="aaaaaa"
```

```
signature="bbbbbb" />
```

### Example 11.70. Add User Credentials: JSON response

```
{
    "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "secret": "aaaaaa",
        "signature": "bbb"
    }
}
```

## 11.4.3. Get User Credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Gets user credentials.

**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)

### 11.4.3.1. 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.

### 11.4.3.2. Response

#### Example 11.71. Get User Credentials: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ec2Credentials
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSEC2/v1.0"
    username="testuser"
    key="aaaaaa"
    signature="bbbbbb" />
```

#### Example 11.72. Get User Credentials: JSON response

```
{
    "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "secret": "aaaaaa",
        "signature": "bbb"
    }
}
```

## 11.4.4. Update User Credentials

Method	URI	Description
POST	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials	Updates credentials for a specified 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)

### 11.4.4.1. 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 11.73. Update User Credentials: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<ec2Credentials
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSEC2/v1.0"
    username="testuser"
    key="aaaaaa"
    signature="bbbbbb" />
```

#### Example 11.74. Update User Credentials: JSON request

```
{
    "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "secret": "aaaaaa",
        "signature": "bbb"
    }
}
```

### 11.4.4.2. Response

#### Example 11.75. Update User Credentials: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ec2Credentials
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSEC2/v1.0"
    username="testuser"
    key="aaaaaa"
```

```
signature="bbbbbb" />
```

### Example 11.76. Update User Credentials: JSON response

```
{
    "OS-KSEC2-ec2Credentials": {
        "username": "test_user",
        "secret": "aaaaaa",
        "signature": "bbb"
    }
}
```

## 11.4.5. 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)

### 11.4.5.1. 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.

## 11.4.6. List Credentials by Type

Method	URI	Description
GET	/v2.0/users/{userId}/OS-KSADM/credentials/OS-KSEC2:ec2Credentials/{type}{?type}	Lists credentials by type.

**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)

### 11.4.6.1. Request

This table shows the header parameters for the list credentials by type request:

Name	Type	Description
X-Auth-Token	String <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 operation does not accept a request body.

### 11.4.6.2. Response

#### Example 11.77. List Credentials by Type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<credentials xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://docs.openstack.org/identity/api/v2.0">
  <passwordCredentials username="test_user" password="test"/>
</credentials>
```

#### Example 11.78. List Credentials by Type: JSON response

```
<?xml version="1.0" encoding="UTF-8"?>
<credentials xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://docs.openstack.org/identity/api/v2.0">
  <passwordCredentials username="test_user" password="test"/>
</credentials>
```

This operation does not return a response body.

## 11.5. OS-KSS3 admin extension

Method	URI	Description
GET	/v2.0/users/{userId}/OS-OS-KSS3/credentials{?marker,limit}	Lists credentials.

Method	URI	Description
<b>POST</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials{?marker,limit}	Adds a credential to a user.
<b>GET</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials/s3credentials	Gets user credentials.
<b>POST</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials/s3credentials	Updates credentials.
<b>DELETE</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials/s3credentials	Deletes user credentials.
<b>GET</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials/s3credentials/{type}{?type}	Lists credentials by type.

## 11.5.1. List credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-OS-KSS3/credentials{?marker,limit}	Lists credentials.

**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)

### 11.5.1.1. 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 operation does not accept a request body.

### 11.5.1.2. Response

#### Example 11.79. List credentials: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<credentials xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://docs.openstack.org/identity/api/v2.0">
  <passwordCredentials username="test_user" password="test"/>
  <s3Credentials xmlns="http://docs.openstack.org/identity/api/ext/OS-KSS3/
v1.0"
    username="testuser" key="aaaaaa" signature="bbbbbb"/>
</credentials>
```

#### Example 11.80. List credentials: JSON response

```
{
  "credentials": [
    {
      "passwordCredentials": {
        "username": "test_user",
        "password": "mypass"
      }
    },
    {
      "OS-KSS3:s3Credentials": {
        "username": "test_user",
        "key": "aaaaaa",
        "signature": "bbbbbb"
      }
    }
  ]
}
```

```
        "secret": "aaaaa",
        "signature": "bbb"
    }
},
"credentials_links": []
}
```

## 11.5.2. Add user credential

Method	URI	Description
POST	/v2.0/users/{userId}/OS-OS-KSS3/credentials{?marker,limit}	Adds 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)

### 11.5.2.1. Request

This table shows the header parameters for the add user credential request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token.

This table shows the URI parameters for the add user credential request:

Name	Type	Description
{userId}	String	The user ID.

#### Example 11.81. Add user credential: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<s3Credentials
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSS3/v1.0"
    username="testuser"
    key="aaaaaa"
    signature="bbbbbb" />
```

#### Example 11.82. Add user credential: JSON request

```
{
    "OS-KSS3:s3Credentials": {
        "username": "test_user",
        "secret": "aaaaaa",
        "signature": "bbb"
    }
}
```

### 11.5.2.2. Response

#### Example 11.83. Add user credential: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<s3Credentials
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSS3/v1.0"
    username="testuser"
    key="aaaaaa"
```

```
signature="bbbbbb" />
```

### Example 11.84. Add user credential: JSON response

```
{
    "OS-KSS3:s3Credentials": {
        "username": "test_user",
        "secret": "aaaaaa",
        "signature": "bbb"
    }
}
```

## 11.5.3. Get user credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-OS-KSS3/credentials/s3credentials	Gets user credentials.

**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)

### 11.5.3.1. 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.

### 11.5.3.2. Response

#### Example 11.85. Get user credentials: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<s3Credentials
  xmlns="http://docs.openstack.org/identity/api/ext/OS-KSS3/v1.0"
  username="testuser"
  key="aaaaaa"
  signature="bbbbbb" />
```

#### Example 11.86. Get user credentials: JSON response

```
{
  "OS-KSS3:s3Credentials": {
    "username": "test_user",
    "secret": "aaaaaa",
    "signature": "bbb"
  }
}
```

## 11.5.4. Update user credentials

Method	URI	Description
POST	/v2.0/users/{userId}/OS-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)

### 11.5.4.1. 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 11.87. Update user credentials: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<s3Credentials
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSS3/v1.0"
    username="testuser"
    key="aaaaaa"
    signature="bbbbbb" />
```

#### Example 11.88. Update user credentials: JSON request

```
{
    "OS-KSS3:s3Credentials": {
        "username": "test_user",
        "secret": "aaaaaa",
        "signature": "bbb"
    }
}
```

### 11.5.4.2. Response

#### Example 11.89. Update user credentials: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<s3Credentials
    xmlns="http://docs.openstack.org/identity/api/ext/OS-KSS3/v1.0"
    username="testuser"
    key="aaaaaa"
```

```
signature="bbbbbb" />
```

### Example 11.90. Update user credentials: JSON response

```
{
    "OS-KSS3:s3Credentials": {
        "username": "test_user",
        "secret": "aaaaaa",
        "signature": "bbb"
    }
}
```

## 11.5.5. Delete user credentials

Method	URI	Description
DELETE	/v2.0/users/{userId}/OS-OS-KSS3/credentials/s3credentials	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)

### 11.5.5.1. 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.

## 11.5.6. List credentials by type

Method	URI	Description
<b>GET</b>	/v2.0/users/{userId}/OS-OS-KSS3/credentials/s3credentials/{type}{?type}	Lists credentials by type.

**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)

### 11.5.6.1. Request

This table shows the header parameters for the list credentials by type request:

Name	Type	Description
X-Auth-Token	String	A valid authentication token. <i>(Required)</i>

This table shows the URI parameters for the list credentials by type request:

Name	Type	Description
{userId}	String	The user ID.

This operation does not accept a request body.

### 11.5.6.2. Response

#### Example 11.91. List credentials by type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<credentials xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://docs.openstack.org/identity/api/v2.0">
  <passwordCredentials username="test_user" password="test"/>
  <s3Credentials xmlns="http://docs.openstack.org/identity/api/ext/OS-KSS3/
v1.0">
    <username="testuser" key="aaaaaa" signature="bbbbbb" />
  </s3Credentials>
```

#### Example 11.92. List credentials by type: JSON response

```
{
  "credentials": [
    {
      "passwordCredentials": {
        "username": "test_user",
        "password": "mypass"
      }
    },
    {
      "OS-KSS3:s3Credentials": {
        "key": "aaaaaa",
        "signature": "bbbbbb"
      }
    }
  ]
}
```

```
        "username": "test_user",
        "secret": "aaaaa",
        "signature": "bbb"
    }
},
],
"credentials_links": []
}
```

## 11.6. OS-KVALIDATE admin extension

Method	URI	Description
<b>GET</b>	/v2.0/OS-KVALIDATE/token/validate {?belongsTo,HP-IDM-serviceId}	Checks that a token is valid and that it belongs to a specified tenant and service IDs. Returns the permissions for a particular client.
<b>HEAD</b>	/v2.0/OS-KVALIDATE/token/validate {?belongsTo,HP-IDM-serviceId}	Checks that a token is valid and that it belongs to a specified tenant and service IDs, for performance.
<b>GET</b>	/v2.0/OS-KVALIDATE/token/endpoints{?HP-IDM-serviceId}	Lists endpoints associated with a specific token.

## 11.6.1. Validate token

Method	URI	Description
<b>GET</b>	/v2.0/OS-KVALIDATE/token/validate {?belongsTo,HP-IDM-serviceId}	Checks that a token is valid and that it belongs to a specified tenant and service IDs. Returns the permissions for a particular client.

Behavior is similar to /tokens/{tokenId}. An itemNotFound (404) fault is returned for a token that is not valid.

This extension might decrypt X-Subject-Token header and internally call the normal validation for Identity, passing in all headers and query parameters. It should therefore support all existing calls on /tokens/{tokenId}, including extensions such as HP-IDM.

**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)

### 11.6.1.1. 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.

### 11.6.1.2. Response

#### Example 11.93. Validate token: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<access xmlns="http://docs.openstack.org/identity/api/v2.0">
    <token id="ab48a9efdfedb23ty3494" expires="2010-11-01T03:32:15-05:00">
        <tenant id="456" name="My Project" />
    </token>
    <user id="123" name="jqsmith">
        <roles xmlns="http://docs.openstack.org/identity/api/v2.0">
            <role id="123" name="Admin" tenantId="one"/>
            <role id="234" name="object-store:admin" tenantId="1"/>
        </roles>
    </user>
</access>
```

#### Example 11.94. Validate token: JSON response

```
{
```

```
"access": {
    "token": {
        "id": "ab48a9efdfedb23ty3494",
        "expires": "2010-11-01T03:32:15-05:00",
        "tenant": {
            "id": "345",
            "name": "My Project"
        }
    },
    "user": {
        "id": "123",
        "name": "jqsmith",
        "roles": [
            {
                "id": "234",
                "name": "compute:admin"
            },
            {
                "id": "234",
                "name": "object-store:admin",
                "tenantId": "1"
            }
        ],
        "roles_links": []
    }
}
```

## 11.6.2. Check token

Method	URI	Description
HEAD	/v2.0/OS-KSVALIDATE/token/validate {?belongsTo,HP-IDM-serviceId}	Checks that a token is valid and that it belongs to a specified tenant and service IDs, for performance.

Behavior is similar to /tokens/{tokenId}. An itemNotFound (404) fault is returned for a token that is not valid.

This extension might decrypt X-Subject-Token header and internally call the normal validation for Identity, passing in all headers and query parameters. It should therefore support all existing calls on /tokens/{tokenId}, including extensions such as HP-IDM.

**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)

### 11.6.2.1. 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.

## 11.6.3. List endpoints for a token

Method	URI	Description
GET	/v2.0/OS-KSVALIDATE/token/endpoints{?HP-IDM-serviceId}	Lists endpoints associated with a specific token.

**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)

### 11.6.3.1. Request

This table shows the header parameters for the list endpoints for a 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.

### 11.6.3.2. Response

#### Example 11.95. List endpoints for a token: XML response

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

<endpoints
    xmlns="http://docs.openstack.org/identity/api/v2.0">
    <endpoint
        id="1"
        tenantId="1"
        type="compute"
        name="Compute"
        region="North"
        publicURL="https://compute.north.public.com/v1"
        internalURL="https://compute.north.internal.com/v1"
        adminURL="https://compute.north.internal.com/v1">
        <version
            id="1"
            info="https://compute.north.public.com/v1/"
            list="https://compute.north.public.com/">
        />
    </endpoint>
    <endpoint
        id="2"
        tenantId="2"
        type="compute"
        name="Compute"
        region="South"
        publicURL="https://compute.north.public.com/v1"
        internalURL="https://compute.north.internal.com/v1">
    </endpoint>
</endpoints>
```

```

adminURL="https://compute.north.internal.com/v1">
<version
    id="1"
    info="https://compute.north.public.com/v1/"
    list="https://compute.north.public.com/"
/>
</endpoint>
<endpoint
    id="3"
    tenantId="1"
    type="compute"
    name="Compute"
    region="East"
    publicURL="https://compute.north.public.com/v1"
    internalURL="https://compute.north.internal.com/v1"
    adminURL="https://compute.north.internal.com/v1"
/>
<endpoint
    id="4"
    tenantId="1"
    type="compute"
    name="Compute"
    region="West"
    publicURL="https://compute.north.public.com/v1"
    internalURL="https://compute.north.internal.com/v1"
    adminURL="https://compute.north.internal.com/v1">
<version
    id="1"
    info="https://compute.north.public.com/v1/"
    list="https://compute.north.public.com/"
/>
</endpoint>
<endpoint
    id="5"
    tenantId="1"
    type="compute"
    name="Compute"
    region="Global"
    publicURL="https://compute.north.public.com/v1"
    internalURL="https://compute.north.internal.com/v1"
    adminURL="https://compute.north.internal.com/v1">
<version
    id="1"
    info="https://compute.north.public.com/v1/"
    list="https://compute.north.public.com/"
/>
</endpoint>
</endpoints>

```

### Example 11.96. List endpoints for a token: JSON response

```
{
  "endpoints": [
    {
      "id": 1,
      "tenantId": "1",
      "region": "North",
      "type": "compute",
      "publicURL": "https://compute.north.public.com/v1",
      "internalURL": "https://compute.north.internal.com/v1",
      "adminURL": "https://compute.north.internal.com/v1"
    }
  ]
}
```

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

## 11.7. RAX-GRPADM admin extension

Method	URI	Description
<b>GET</b>	/v2.0/RAX-GRPADM/groups{?marker,limit,name}	Lists groups.
<b>POST</b>	/v2.0/RAX-GRPADM/groups{?marker,limit,name}	Adds a group.
<b>GET</b>	/v2.0/RAX-GRPADM/groups/{groupId}	Gets information for a group by ID.
<b>PUT</b>	/v2.0/RAX-GRPADM/groups/{groupId}	Updates a group.
<b>DELETE</b>	/v2.0/RAX-GRPADM/groups/{groupId}	Deletes a group.
<b>GET</b>	/v2.0/RAX-GRPADM/groups/{groupId}/users{?marker,limit}	Lists users for a group.
<b>PUT</b>	/v2.0/RAX-GRPADM/groups/{groupId}/users/{userId}	Adds a user to a group.
<b>DELETE</b>	/v2.0/RAX-GRPADM/groups/{groupId}/users/{userId}	Removes a user from a group.

## 11.7.1. List groups

Method	URI	Description
GET	/v2.0/RAX-GRPADM/groups{?marker,limit,name}	Lists groups.

**Normal response codes:** 200, 203

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

### 11.7.1.1. 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 operation does not accept a request body.

### 11.7.1.2. Response

This table shows the body parameters for the list groups response:

Name	Type	Description
next	AnyURI <i>(Optional)</i>	
previous	AnyURI <i>(Optional)</i>	

#### Example 11.97. List groups: XML response

```
<groups xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSGRP/v1.0"
       xmlns:atom="http://www.w3.org/2005/Atom">
    <group id="1234" name="group1">
        <description>A Description of the group</description>
    </group>
    <group id="1235" name="group2">
        <description>A Description of the group</description>
    </group>
    <group id="1236" name="group3">
        <description>A Description of the group</description>
    </group>
    <atom:link rel="next" href="https://identity.openstack.com/v2.0/RAX-
GRPADM/groups?marker=1236"/>
</groups>
```

#### Example 11.98. List groups: JSON response

```
{
    "RAX-KSGRP:groups": [
```

```
{  
    "id": "1234",  
    "name": "group1",  
    "description": "A Description of the group"  
},  
{  
    "id": "1235",  
    "name": "group2",  
    "description": "A Description of the group"  
},  
{  
    "id": "1236",  
    "name": "group3",  
    "description": "A Description of the group"  
}  
,  
"RAX-KSGRP:groups_links": [  
    {  
        "rel": "next",  
        "href": "https://identity.openstack.com/v2.0/RAX-GRPADM/groups?  
marker=1236"  
    }  
]  
}
```

## 11.7.2. Add group

Method	URI	Description
POST	/v2.0/RAX-GRPADM/groups{ ?marker , limit , name }	Adds a group.

**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)

### 11.7.2.1. Request

This table shows the header parameters for the add group request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

This table shows the body parameters for the add group request:

Name	Type	Description
group	GroupFor Create <i>(Required)</i>	

#### Example 11.99. Add group: XML request

```
<group name="group1" xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSGRP/v1.0">
    <description>A Description of the group</description>
</group>
```

#### Example 11.100. Add group: JSON request

```
{
    "RAX-KSGRP:group": {
        "name": "group1",
        "description": "A Description of the group"
    }
}
```

### 11.7.2.2. Response

#### Example 11.101. Add group: XML response

```
<group id="1234" name="group1" xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSGRP/v1.0">
    <description>A Description of the group</description>
</group>
```

**Example 11.102. Add group: JSON response**

```
{  
    "RAX-KSGRP:group": {  
        "id": "1234",  
        "name": "group1",  
        "description": "A Description of the group"  
    }  
}
```

## 11.7.3. Get group information

Method	URI	Description
GET	/v2.0/RAX-GRPADM/groups/{groupId}	Gets information for a group by ID.

**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)

### 11.7.3.1. Request

This table shows the header parameters for the get group information 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 group information request:

Name	Type	Description
{groupId}	String	The group ID.

This operation does not accept a request body.

### 11.7.3.2. Response

#### Example 11.103. Get group information: XML response

```
<group id="1234" name="group1" xmlns="http://docs.rackspace.com/identity/api/
ext/RAX-KSGRP/v1.0">
    <description>A Description of the group</description>
</group>
```

#### Example 11.104. Get group information: JSON response

```
{
    "RAX-KSGRP:group": {
        "id": "1234",
        "name": "group1",
        "description": "A Description of the group"
    }
}
```

## 11.7.4. Update group

Method	URI	Description
PUT	/v2.0/RAX-GRPADM/groups/{groupId}	Updates a group.

**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)

### 11.7.4.1. 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
{groupId}	String	The group ID.

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

Name	Type	Description
group	GroupForUpdate <i>(Required)</i>	

#### Example 11.105. Update group: XML request

```
<group name="newName" xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSGRP/v1.0">
    <description>A new description</description>
</group>
```

#### Example 11.106. Update group: JSON request

```
{
    "RAX-KSGRP:group": {
        "name": "newName",
        "description": "A Description of the group"
    }
}
```

### 11.7.4.2. Response

#### Example 11.107. Update group: XML response

```
<group id="1234" name="newName" xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSGRP/v1.0">
```

```
<description>A new description</description>
</group>
```

### Example 11.108. Update group: JSON response

```
{
    "RAX-KSGRP:group": {
        "id": "1234",
        "name": "newName",
        "description": "A new description"
    }
}
```

## 11.7.5. Delete group

Method	URI	Description
DELETE	/v2.0/RAX-GRPADM/groups/{groupId}	Deletes a group.

**Normal response codes:** 204

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

### 11.7.5.1. 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
{groupId}	String	The group ID.

This operation does not accept a request body.

## 11.7.6. List users for group

Method	URI	Description
GET	/v2.0/RAX-GRPADM/groups/{groupId}/users{?marker,limit}	Lists users for a group.

**Normal response codes:** 200, 203

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

### 11.7.6.1. Request

This table shows the header parameters for the list users for 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 for group request:

Name	Type	Description
{groupId}	String	The group ID.

This table shows the query parameters for the list users for group request:

Name	Type	Description
marker	String <i>(Optional)</i>	
limit	Int <i>(Optional)</i>	

This operation does not accept a request body.

### 11.7.6.2. Response

This table shows the body parameters for the list users for group response:

Name	Type	Description
next	AnyURI <i>(Optional)</i>	
previous	AnyURI <i>(Optional)</i>	

#### Example 11.109. List users for group: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<users xmlns="http://docs.openstack.org/identity/api/v2.0">
    <user xmlns="http://docs.openstack.org/identity/api/v2.0">
```

```
        enabled="true" email="john.smith@example.org"
        username="jqsmith" id="u1000"/>
<user xmlns="http://docs.openstack.org/identity/api/v2.0"
        enabled="true" email="john.smith@example.org"
        username="jqsmith" id="u1001"/>
</users>
```

### Example 11.110. List users for group: JSON response

```
{
    "users": [
        {
            "id": "u1000",
            "username": "jqsmith",
            "email": "john.smith@example.org",
            "enabled": true
        },
        {
            "id": "u1001",
            "username": "jqsmith",
            "email": "john.smith@example.org",
            "enabled": true
        }
    ],
    "users_links": []
}
```

## 11.7.7. Add user to group

Method	URI	Description
PUT	/v2.0/RAX-GRPADM/groups/{groupId}/users/{userId}	Adds a user to a group.

**Normal response codes:** 204

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

### 11.7.7.1. Request

This table shows the header parameters for the add 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 add user to group request:

Name	Type	Description
{groupId}	String	The group ID.
{userId}	String	The user ID.

This operation does not accept a request body.

## 11.7.8. Remove user from group

Method	URI	Description
<b>DELETE</b>	/v2.0/RAX-GRPADM/groups/{groupId}/users/{userId}	Removes a user from a group.

**Normal response codes:** 204

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

### 11.7.8.1. 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
{groupId}	String	The group ID.
{userId}	String	The user ID.

This operation does not accept a request body.

## 11.8. RAX-KSGRP admin extension

Method	URI	Description
<b>GET</b>	/v2.0/users/{userId}/RAX-KSGRP	Lists groups for a user.

## 11.8.1. List groups for a user

Method	URI	Description
GET	/v2.0/users/{userId}/RAX-KSGRP	Lists groups for a user.

**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)

### 11.8.1.1. Request

This table shows the header parameters for the list groups for a 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 a user request:

Name	Type	Description
{userId}	String	The user ID.

This operation does not accept a request body.

### 11.8.1.2. Response

#### Example 11.111. List groups for a user: XML response

```
<groups xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSGRP/v1.0">
  <group xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSGRP/v1.0" id="1" name="Default" >
    <description>Default Limits</description>
  </group>
  <group xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSGRP/v1.0" id="1550" name="New Group 1" >
    <description>This is the first new group.</description>
  </group>
  <group xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSGRP/v1.0" id="214" name="Faster Defaults" >
    <description>Defaults with faster rate limits</description>
  </group>
</groups>
```

#### Example 11.112. List groups for a user: JSON response

```
{
  "RAX-KSGRP:groups": [
    {
      "description": "Default Limits",
      "id": "1",
      "name": "Default"
```

```
        },
        {
            "description": "This is the first new group.",
            "id": "1550",
            "name": "New Group 1"
        },
        {
            "description": "Defaults with faster rate limits",
            "id": "214",
            "name": "Faster Defaults"
        }
    ],
    "RAX-KSGRP:groups_links": []
}
```

## 11.9. RAX-KSKEY admin extension

Method	URI	Description
<b>POST</b>	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials	Adds a credential to a user.
<b>GET</b>	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials/{?marker,limit}	Lists credentials.
<b>POST</b>	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials/RAX-KSKEY:apiKeyCredentials	Updates credentials.
<b>DELETE</b>	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials/RAX-KSKEY:apiKeyCredentials	Deletes user credentials.
<b>GET</b>	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials/RAX-KSKEY:apiKeyCredentials	Gets user credentials.

## 11.9.1. Add user credential

Method	URI	Description
POST	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials	Adds 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)

### 11.9.1.1. Request

This table shows the header parameters for the add user 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 add user credential request:

Name	Type	Description
{userId}	String	The user ID.

#### Example 11.113. Add user credential: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<apiKeyCredentials
    xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSKEY/v1.0"
    username="testuser"
    apiKey="aaaaaa-bbbb-bcccc-12345678" />
```

#### Example 11.114. Add user credential: JSON request

```
{
    "RAX-KSKEY:apiKeyCredentials": {
        "username": "test_user",
        "apiKey": "aaaaaa-bbbb-bcccc-12345678"
    }
}
```

### 11.9.1.2. Response

#### Example 11.115. Add user credential: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<apiKeyCredentials
    xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSKEY/v1.0"
    username="testuser"
    apiKey="aaaaaa-bbbb-bcccc-12345678" />
```

**Example 11.116. Add user credential: JSON response**

```
{  
    "RAX-KSKEY:apiKeyCredentials": {  
        "username": "test_user",  
        "apiKey": "aaaaaa-bbbb-bcccc-12345678"  
    }  
}
```

## 11.9.2. List credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials/{?marker,limit}	Lists credentials.

**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)

### 11.9.2.1. 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.
X-Auth-Token	String <i>(Required)</i>	A valid authentication token for an administrative user.

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

Name	Type	Description
{userId}	String	The user ID.

This operation does not accept a request body.

### 11.9.2.2. Response

#### Example 11.117. List credentials: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<credentials xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://docs.openstack.org/identity/api/v2.0">
  <passwordCredentials username="test_user" password="test"/>
  <apiKeyCredentials
    xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSKEY/v1.0"
    username="testuser"
    apiKey="aaaaaa-bbbb-bccc-cccc-12345678"/>
</credentials>
```

#### Example 11.118. List credentials: JSON response

```
{
  "credentials": [
    {
      "passwordCredentials": {
        "username": "test_user",
        "password": "mypass"
      }
    }
  ]
}
```

```
        },
        {
            "RAX-KSKEY:apiKeyCredentials": {
                "username": "test_user",
                "apiKey": "aaaaaa-bbbbb-cccccc-12345678"
            }
        }
    ],
    "credentials_links": []
}
```

## 11.9.3. Update user credentials

Method	URI	Description
POST	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials/RAX-KSKEY:apiKeyCredentials	Updates credentials.

**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)

### 11.9.3.1. 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 for an administrative user.

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

Name	Type	Description
{userId}	String	The user ID.

#### Example 11.119. Update user credentials: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<apiKeyCredentials
    xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSKEY/v1.0"
    username="testuser"
    apiKey="aaaaaa-bbbb-bcccc-12345678" />
```

#### Example 11.120. Update user credentials: JSON request

```
{
    "RAX-KSKEY:apiKeyCredentials": {
        "username": "test_user",
        "apiKey": "aaaaaa-bbbb-bcccc-12345678"
    }
}
```

### 11.9.3.2. Response

#### Example 11.121. Update user credentials: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<apiKeyCredentials
    xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSKEY/v1.0"
    username="testuser"
    apiKey="aaaaaa-bbbb-bcccc-12345678" />
```

**Example 11.122. Update user credentials: JSON response**

```
{  
    "RAX-KSKEY:apiKeyCredentials": {  
        "username": "test_user",  
        "apiKey": "aaaaaa-bbbb-bcccc-12345678"  
    }  
}
```

## 11.9.4. Delete user credentials

Method	URI	Description
<b>DELETE</b>	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials/RAX-KSKEY:apiKeyCredentials	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)

### 11.9.4.1. 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 for an administrative user.

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.

## 11.9.5. Get user credentials

Method	URI	Description
GET	/v2.0/users/{userId}/OS-RAX-KSKEY/credentials/RAX-KSKEY:apiKeyCredentials	Gets user credentials.

**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)

### 11.9.5.1. 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 for an administrative user.

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.

### 11.9.5.2. Response

#### Example 11.123. Get user credentials: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<apiKeyCredentials
    xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSKEY/v1.0"
    username="testuser"
    apiKey="aaaaaa-bbbb-bcccc-12345678"/>
```

#### Example 11.124. Get user credentials: JSON response

```
{
    "RAX-KSKEY:apiKeyCredentials": {
        "username": "test_user",
        "apiKey": "aaaaaa-bbbb-bcccc-12345678"
    }
}
```

## 11.10. RAX-KSQA admin extension

Method	URI	Description
GET	/v2.0/users/{userId}/RAX-KSQA/se-cretqa	Gets a secret question and answer for a specified user.

Method	URI	Description
<b>PUT</b>	/v2.0/users/{userId}/RAX-KSQA/se-cretqa	Updates a secret question and answer for a specified user.

## 11.10.1. Get user secret question and answer

Method	URI	Description
GET	/v2.0/users/{userId}/RAX-KSQA/secretqa	Gets a secret question and answer for a specified user.

**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)

### 11.10.1.1. Request

This table shows the header parameters for the get user secret question and answer 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 user secret question and answer request:

Name	Type	Description
{userId}	String	The user ID.

This operation does not accept a request body.

### 11.10.1.2. Response

#### Example 11.125. Get user secret question and answer: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<secretQA xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSQA/v1.0"
    question="What is the color of my eyes?"
    answer="Leonardo da Vinci" />
```

#### Example 11.126. Get user secret question and answer: JSON response

```
{
    "RAX-KSQA:secretQA": {
        "question": "What is the color of my eyes?",
        "answer": "Leonardo da Vinci"
    }
}
```

## 11.10.2. Update user secret question and answer

Method	URI	Description
PUT	/v2.0/users/{userId}/RAX-KSQA/se-cretqa	Updates a secret question and answer for a specified 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)

### 11.10.2.1. Request

This table shows the header parameters for the update user secret question and answer 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 secret question and answer request:

Name	Type	Description
{userId}	String	The user ID.

#### Example 11.127. Update user secret question and answer: XML request

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

<secretQA xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSQA/v1.0"
    question="What is the color of my eyes?"
    answer="Leonardo da Vinci" />
```

#### Example 11.128. Update user secret question and answer: JSON request

```
{
    "RAX-KSQA:secretQA": {
        "question": "What is the color of my eyes?",
        "answer": "Leonardo da Vinci"
    }
}
```

### 11.10.2.2. Response

#### Example 11.129. Update user secret question and answer: XML response

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

<secretQA xmlns="http://docs.rackspace.com/identity/api/ext/RAX-KSQA/v1.0"
    question="What is the color of my eyes?"
    answer="Leonardo da Vinci" />
```

**Example 11.130. Update user secret question and answer: JSON response**

```
{  
    "RAX-KSQA:secretQA": {  
        "question": "What is the color of my eyes?",  
        "answer": "Leonardo da Vinci"  
    }  
}
```

# 12. Image Service API v2 (SUPPORTED)

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*.

## 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": "CURRENT",
            "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

Create, update, and delete image metadata records. Enable users to share images with each other. Also, upload and download raw image data.

Method	URI	Description
<b>POST</b>	/v2/images	Creates a virtual machine (VM) image. (Since Image API v2.0.)
<b>GET</b>	/v2/images{?limit,marker,name,visibility,member_status,owner,status,size_min,size_max,sort_key,sort_dir,tag}	Lists public virtual machine (VM) images. (Since Image API v2.0.)
<b>GET</b>	/v2/images/{image_id}	Gets details for a specified image. (Since Image API v2.0.)
<b>PATCH</b>	/v2/images/{image_id}	Updates a specified image. (Since Image API v2.0.)
<b>DELETE</b>	/v2/images/{image_id}	Deletes a specified image. (Since Image API v2.0.)

## 12.2.1. Create image

Method	URI	Description
POST	/v2/images	Creates a virtual machine (VM) image. (Since Image API v2.0.)

Created with a `Location` header that contains the newly-created URI for the image. Response body represents the created image entity.

**Normal response codes:** 201

### 12.2.1.1. Request

#### Example 12.2. Create image: JSON request

```
{  
    "id": "e7db3b45-8db7-47ad-8109-3fb55c2c24fd",  
    "name": "Ubuntu 12.10",  
    "tags": [  
        "ubuntu",  
        "quantal"  
    ]  
}
```

### 12.2.1.2. Response

#### Example 12.3. Create image: JSON response

```
{  
    "image": {  
        "status": "queued",  
        "deleted": false,  
        "container_format": null,  
        "min_ram": 0,  
        "updated_at": "2014-05-05T21:15:26",  
        "owner": "f3490b9d3efd47df9e6e91845a1d647b",  
        "min_disk": 0,  
        "is_public": false,  
        "deleted_at": null,  
        "id": "e7db3b45-8db7-47ad-8109-3fb55c2c24fd",  
        "size": 0,  
        "virtual_size": null,  
        "name": "Ubuntu 12.10",  
        "checksum": null,  
        "created_at": "2014-05-05T21:15:26",  
        "disk_format": null,  
        "properties": {},  
        "protected": false  
    }  
}
```

## 12.2.2. List images

Method	URI	Description
<b>GET</b>	/v2/images{?limit,marker,name,visibility,member_status,owner,status,size_min,size_max,sort_key,sort_dir,tag}	Lists public virtual machine (VM) images. (Since Image API v2.0.)

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 sort the results of this operation by using the `sort_key` and `sort_dir` parameters. The API uses the natural sorting of whatever image attribute is provided as the `sort_key`.

**Normal response codes:** 200

### 12.2.2.1. Request

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

Name	Type	Description
<code>limit</code>	<code>DateTime</code> <i>(Optional)</i>	Use to request a specific page size. Expect a response to a limited request to return between zero and limit items. The typical pattern of <code>limit</code> and <code>marker</code> is to make an initial limited request and then to use the ID of the last image from the response as the <code>marker</code> parameter in a subsequent limited request.
<code>marker</code>	<code>String</code> <i>(Optional)</i>	Specifies the ID of the last-seen image. The typical pattern of <code>limit</code> and <code>marker</code> is to make an initial limited request and then to use the ID of the last image from the response as the <code>marker</code> parameter in a subsequent limited request.
<code>name</code>	<code>String</code> <i>(Optional)</i>	Filter parameter. Name of the image as a string.

Name	Type	Description
visibility	String <i>(Optional)</i>	Filter parameter. Image visibility. Valid values are public, private, and shared. Default is public, private, and shared images with accepted member status.
member_status	String <i>(Optional)</i>	Filter parameter. Only show images with the specified member status. Valid values are accepted, pending, rejected, and all. Default is accepted.
owner	String <i>(Optional)</i>	Filter parameter. Shows images shared with me by the specified owner, where the owner is indicated by tenant ID.
status	Int <i>(Optional)</i>	Filter parameter. The image status, such as queued, saving, active, killed, deleted, and pending_delete.
size_min	String <i>(Optional)</i>	Filter parameter. Value of the minimum size of the image in bytes.
size_max	String <i>(Optional)</i>	Filter parameter. Value of the maximum size of the image in bytes.
sort_key	String <i>(Optional)</i>	Sort key. All image attributes can be used as the sort_key, except tags and link attributes. Default is created_at.
sort_dir	String <i>(Optional)</i>	Sort direction. Valid values are asc (ascending) and desc (descending). Default is desc.
tag	String <i>(Optional)</i>	Image tag. For example, ?tag="cirros".

This operation does not accept a request body.

### 12.2.2.2. Response

#### Example 12.4. List images: JSON response

```
{
  "images": [
    {
      "status": "active",
      "deleted_at": null,
      "name": "cirros-0.3.2-x86_64-disk",
      "deleted": false,
      "container_format": "bare",
      "created_at": "2014-05-05T17:15:10",
      "disk_format": "qcow2",
      "updated_at": "2014-05-05T17:15:11",
      "min_disk": 0,
      "protected": false,
      "id": "1bea47ed-f6a9-463b-b423-14b9cca9ad27",
      "min_ram": 0,
      "checksum": "64d7c1cd2b6f60c92c14662941cb7913",
      "owner": "5ef70662f8b34079a6eddb8da9d75fe8",
      "is_public": true,
      "virtual_size": null,
      "properties": {},
      "size": 13167616
    },
    {
      "status": "active",
      "deleted_at": null,
      "name": "cirros-0.3.2-x86_64-disk",
      "deleted": false,
      "container_format": "bare",
      "created_at": "2014-05-05T17:15:10",
      "disk_format": "qcow2",
      "updated_at": "2014-05-05T17:15:11",
      "min_disk": 0,
      "protected": false,
      "id": "1bea47ed-f6a9-463b-b423-14b9cca9ad27",
      "min_ram": 0,
      "checksum": "64d7c1cd2b6f60c92c14662941cb7913",
      "owner": "5ef70662f8b34079a6eddb8da9d75fe8",
      "is_public": true,
      "virtual_size": null,
      "properties": {},
      "size": 13167616
    }
  ]
}
```

```
    "deleted_at": null,
    "name": "F17-x86_64-cfntools",
    "deleted": false,
    "container_format": "bare",
    "created_at": "2014-05-05T17:15:11",
    "disk_format": "qcow2",
    "updated_at": "2014-05-05T17:15:16",
    "min_disk": 0,
    "protected": false,
    "id": "781b3762-9469-4cec-b58d-3349e5de4e9c",
    "min_ram": 0,
    "checksum": "afab0f79bac770d61d24b4d0560b5f70",
    "owner": "5ef70662f8b34079a6eddb8da9d75fe8",
    "is_public": true,
    "virtual_size": null,
    "properties": {},
    "size": 476704768
  }
]
}
```

## 12.2.3. Get image details

Method	URI	Description
GET	/v2/images/{image_id}	Gets details for a specified image. (Since Image API v2.0.)

Response body is a single image entity.

**Normal response codes:** 200

### 12.2.3.1. Request

This operation does not accept a request body.

### 12.2.3.2. Response

#### Example 12.5. Get 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"  
}
```

## 12.2.4. Update image

Method	URI	Description
PATCH	/v2/images/{image_id}	Updates a specified image. (Since Image API v2.0.)

Request body must conform to one of the following 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 [Appendix B: HTTP PATCH media types](#) in the *OpenStack Image Service API v2 Reference*.

**Normal response codes:** 200

### 12.2.4.1. Request

#### Example 12.6. Update image: JSON request

```
[
  {
    "op": "replace",
    "path": "/name",
    "value": "Fedora 17"
  },
  {
    "op": "replace",
    "path": "/tags",
    "value": [
      "fedora",
      "beefy"
    ]
  }
]
```

### 12.2.4.2. Response

#### Example 12.7. 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"
}
```

## 12.2.5. Delete image

Method	URI	Description
<b>DELETE</b>	/v2/images/{image_id}	Deletes a specified image. (Since Image API v2.0.)

You cannot delete images 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) and then perform the delete. The response is empty and returns the HTTP 204 status code.

**Normal response codes:** 204

**Error response codes:** 403

### 12.2.5.1. Request

This operation does not accept a request body.

## 12.3. Image data

Upload and download raw image data.

Method	URI	Description
<b>PUT</b>	/v2/images/{image_id}/file	Uploads binary image data. (Since Image API v2.0.)
<b>GET</b>	/v2/images/{image_id}/file	Downloads binary image data. (Since Image API v2.0.)

## 12.3.1. Upload binary image data

Method	URI	Description
PUT	/v2/images/{image_id}/file	Uploads binary image data. (Since Image API v2.0.)

An image record must exist before a client can store binary image data with it.

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

**Normal response codes:** 204

### 12.3.1.1. Request

This operation does not accept a request body.

## 12.3.2. Download binary image data

Method	URI	Description
GET	/v2/images/{image_id}/file	Downloads binary image data. (Since Image API v2.0.)

**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 value application/octet-stream. The Content-MD5 header contains an MD5 checksum of the image data. Clients are encouraged to verify the integrity of the image data they receive by using this checksum.

If no image data exists, returns the HTTP 204 status code.

**Normal response codes:** 200, 204

### 12.3.2.1. Request

This operation does not accept a request body.

## 12.4. Image tags

Add and delete image tags.

Method	URI	Description
PUT	/v2/images/{image_id}/tags/{tag}	Adds a specified tag to a specified image. (Since Image API v2.0.)
DELETE	/v2/images/{image_id}/tags/{tag}	Deletes a specified tag from a specified image. (Since Image API v2.0.)

## 12.4.1. Add image tag

Method	URI	Description
PUT	/v2/images/{image_id}/tags/{tag}	Adds a specified tag to a specified image. (Since Image API v2.0.)

**Normal response codes:** 204

### 12.4.1.1. Request

This table shows the URI parameters for the add image tag request:

Name	Type	Description
{tag}	String	Image tag.
{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.

## 12.4.2. Delete image tag

Method	URI	Description
<b>DELETE</b>	/v2/images/{image_id}/tags/{tag}	Deletes a specified tag from a specified image. (Since Image API v2.0.)

**Normal response codes:** 204

### 12.4.2.1. Request

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

Name	Type	Description
{tag}	String	Image tag.
{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.

## 12.5. Members

Method	URI	Description
<b>POST</b>	/v2/images/{image_id}/members	Adds a specified tenant ID as an image member. (Since Image API v2.1.)
<b>GET</b>	/v2/images/{image_id}/members	Lists the tenants with whom this image has been shared. (Since Image API v2.1.)
<b>GET</b>	/v2/images/{image_id}/members/{member_id}	Shows image member details.
<b>DELETE</b>	/v2/images/{image_id}/members/{member_id}	Deletes a specified tenant ID from the member list of the specified image. (Since Image API v2.1.)
<b>PUT</b>	/v2/images/{image_id}/members/{member_id}	Sets the specified status for the specified member of the specified image. (Since Image API v2.1.)

## 12.5.1. Create image member

Method	URI	Description
POST	/v2/images/{image_id}/members	Adds a specified tenant ID as an image member. (Since Image API v2.1.)

Normal response codes: 200

### 12.5.1.1. Request

#### Example 12.8. Create image member: JSON request

```
{  
    "member": "8989447062e04a818baf9e073fd04fa7"  
}
```

### 12.5.1.2. Response

#### Example 12.9. Create image member: JSON response

```
{  
    "created_at": "2013-09-20T19:22:19Z",  
    "image_id": "a96be1e-8536-4910-92cb-de50aa19dfe6",  
    "member_id": "8989447062e04a818baf9e073fd04fa7",  
    "schema": "/v2/schemas/member",  
    "status": "pending",  
    "updated_at": "2013-09-20T19:25:31Z"  
}
```

## 12.5.2. List image members

Method	URI	Description
<b>GET</b>	/v2/images/{image_id}/members	Lists the tenants with whom this image has been shared. (Since Image API v2.1.)

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.

**Normal response codes:** 200

**Error response codes:** 404

### 12.5.2.1. Request

This operation does not accept a request body.

### 12.5.2.2. Response

#### Example 12.10. List image members: JSON response

```
{
  "members": [
    {
      "created_at": "2013-10-07T17:58:03Z",
      "image_id": "dbc999e3-c52f-4200-bedd-3b18fe7f87fe",
      "member_id": "123456789",
      "schema": "/v2/schemas/member",
      "status": "pending",
      "updated_at": "2013-10-07T17:58:03Z"
    },
    {
      "created_at": "2013-10-07T17:58:55Z",
      "image_id": "dbc999e3-c52f-4200-bedd-3b18fe7f87fe",
      "member_id": "987654321",
      "schema": "/v2/schemas/member",
      "status": "accepted",
      "updated_at": "2013-10-08T12:08:55Z"
    }
  ],
  "schema": "/v2/schemas/members"
}
```

## 12.5.3. Show image member details

Method	URI	Description
GET	/v2/images/{image_id}/members/{member_id}	Shows image member details.

Response body is a single image member entity. (Since Image API 2.2.)

**Normal response codes:** 200

### 12.5.3.1. Request

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

Name	Type	Description
{member_id}	String	Image member ID. For example, the tenant ID of the user with whom the image is being shared.
{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.

### 12.5.3.2. Response

#### Example 12.11. Show image member details: JSON response

```
{
    "status": "pending",
    "created_at": "2013-11-26T07:21:21Z",
    "updated_at": "2013-11-26T07:21:21Z",
    "image_id": "0ae74cc5-5147-4239-9ce2-b0c580f7067e",
    "member_id": "8989447062e04a818baf9e073fd04fa7",
    "schema": "/v2/schemas/member"
}
```

## 12.5.4. Delete image member

Method	URI	Description
<b>DELETE</b>	/v2/images/{image_id}/members/{member_id}	Deletes a specified tenant ID from the member list of the specified image. (Since Image API v2.1.)

**Normal response codes:** 204

### 12.5.4.1. Request

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

Name	Type	Description
{member_id}	String	Image member ID. For example, the tenant ID of the user with whom the image is being shared.
{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.

## 12.5.5. Update image member

Method	URI	Description
PUT	/v2/images/{image_id}/members/{member_id}	Sets the specified status for the specified member of the specified image. (Since Image API v2.1.)

Normal response codes: 200

### 12.5.5.1. Request

This table shows the URI parameters for the update image member request:

Name	Type	Description
{member_id}	String	Image member ID. For example, the tenant ID of the user with whom the image is being shared.
{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 12.12. Update image member: JSON request

```
{
    "status": "accepted"
}
```

### 12.5.5.2. Response

#### Example 12.13. Update image member: JSON response

```
{
    "created_at": "2013-09-20T19:22:19Z",
    "image_id": "a96be11e-8536-4910-92cb-de50aa19dfe6",
    "member_id": "8989447062e04a818baf9e073fd04fa7",
    "schema": "/v2/schemas/member",
    "status": "accepted",
    "updated_at": "2013-09-20T20:15:31Z"
}
```

## 12.6. Image schemas

Get 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.)

## 12.6.1. 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

### 12.6.1.1. Request

This operation does not accept a request body.

### 12.6.1.2. Response

#### Example 12.14. Get images schema: JSON response

```
{
    "name": "images",
    "properties": {
        "first": {
            "type": "string"
        },
        "images": {
            "items": {
                "name": "image",
                "properties": {
                    "architecture": {
                        "description": "Operating system architecture as specified in http://docs.openstack.org/trunk/openstack-compute/admin/content/adding-images.html",
                        "type": "string"
                    },
                    "checksum": {
                        "description": "md5 hash of image contents. (READ-ONLY)",
                        "maxLength": 32,
                        "type": "string"
                    },
                    "container_format": {
                        "description": "Format of the container",
                        "enum": [
                            "ami",
                            "ari",
                            "aki",
                            "bare",
                            "ovf"
                        ],
                        "type": "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": [
            "ami",
            "ari",
            "aki",
            "vhd",
            "vmdk",
            "raw",
            "qcow2",
            "vdi",
            "iso"
        ],
        "type": "string"
    },
    "file": {
        "description": "(READ-ONLY)",
        "type": "string"
    },
    "id": {
        "description": "An identifier for the image",
        "pattern": "^( [0-9a-fA-F] ) { 8 } - ( [0-9a-fA-F] ) { 4 } - ( [0-9a-
fA-F] ) { 4 } - ( [0-9a-fA-F] ) { 4 } - ( [0-9a-fA-F] ) { 12 } $",
        "type": "string"
    },
    "instance_uuid": {
        "description": "ID of instance used to create this
image.",
        "type": "string"
    },
    "kernel_id": {
        "description": "ID of image stored in Glance that
should be used as the kernel when booting an AMI-style 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"
    },
    "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": "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",
    "type": "string"
},
"os_version": {
    "description": "Operating system version as specified
by the distributor",
    "type": "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.",
    "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": "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"
},
"visibility": {
    "description": "Scope of image accessibility",
    "enum": [
        "public",
        "private"
    ],
    "type": "string"
}
},
"additionalProperties": {
    "type": "string"
},
"links": [
    {
        "href": "{self}",
        "rel": "self"
    },
    {
        "href": "{file}",
        "rel": "enclosure"
    },
    {
        "href": "{schema}",
        "rel": "describedby"
    }
]
},
"type": "array"
},
"next": {
    "type": "string"
},
"schema": {
    "type": "string"
}
},
"links": [
    {
        "href": "{first}"
    }
]
```

```
        "rel": "first"
    },
    {
        "href": "{next}",
        "rel": "next"
    },
    {
        "href": "{schema}",
        "rel": "describedby"
    }
]
```

## 12.6.2. 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

## 12.6.2.1. Request

This operation does not accept a request body.

## 12.6.2.2. Response

### Example 12.15. Get image schema: JSON response

```
        "ari",
        "aki",
        "vhd",
        "vmdk",
        "raw",
        "qcow2",
        "vdi",
        "iso"
    ],
    "type": "string"
},
"file": {
    "description": "(READ-ONLY)",
    "type": "string"
},
"id": {
    "description": "An identifier for the image",
    "pattern": "^( [0-9a-fA-F] {8} - [0-9a-fA-F] {4} - [0-9a-fA-F] {4} - ([0-9a-fA-F] {4} - [0-9a-fA-F] {12}) $",
    "type": "string"
},
"instance_uuid": {
    "description": "ID of instance used to create this image.",
    "type": "string"
},
"kernel_id": {
    "description": "ID of image stored in Glance that should be used as the kernel when booting an AMI-style 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"
},
"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": "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",
        "type": "string"
    },
    "os_version": {
        "description": "Operating system version as specified by the distributor",
        "type": "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.",
        "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": "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"
        },
        "visibility": {
            "description": "Scope of image accessibility",
            "enum": [
                "public",
                "private"
            ],
            "type": "string"
        }
    },
    "additionalProperties": {
        "type": "string"
    },
    "links": [
        {
            "href": "{self}",
            "rel": "self"
        },
        {
            "href": "{file}",
            "rel": "enclosure"
        },
        {
            "href": "{schema}",
            "rel": "describedby"
        }
    ]
}
```

## 12.6.3. 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

### 12.6.3.1. Request

This operation does not accept a request body.

### 12.6.3.2. Response

#### Example 12.16. Get image members schema: JSON response

```
{
    "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"
                    },
                    "status": {
                        "description": "The status of this image member",
                        "enum": [
                            "pending",
                            "accepted",
                            "rejected"
                        ],
                        "type": "string"
                    },
                    "updated_at": {
                        "description": "Date and time of last modification of image member",
                        "type": "string"
                    }
                }
            }
        }
    }
}
```

```
        "type": "string"
    },
    "schema": {
        "type": "string"
    }
}
},
"type": "array"
},
"schema": {
    "type": "string"
}
},
"links": [
{
    "href": "{schema}",
    "rel": "describedby"
}
]
}
```

## 12.6.4. 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

### 12.6.4.1. Request

This operation does not accept a request body.

### 12.6.4.2. Response

#### Example 12.17. 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} ) {12}$",
            "type": "string"
        },
        "member_id": {
            "description": "An identifier for the image member (tenantId)",
            "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"
        },
        "schema": {
            "type": "string"
        }
    }
}
```

# 13. Image Service API v1 (SUPPORTED)

Load images for use at launch time by the Compute API. Also, assign 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		
<b>GET</b>	/	Lists information about all Image Service API versions.
Images		
<b>POST</b>	/v1/{tenant_id}/images	Registers a virtual machine (VM) image.
<b>GET</b>	/v1/{tenant_id}/images/detail{?name,container_format,disk_format,status,size_min,size_max,changes-since}	Lists details for available images.
<b>GET</b>	/v1/{tenant_id}/images/{image_id}	Shows the image details as headers and the image binary in the body of the response.
<b>PUT</b>	/v1/{tenant_id}/images/{image_id}	Updates an image, uploads an image file, or updates metadata for an image.
<b>DELETE</b>	/v1/{tenant_id}/images/{image_id}	Deletes the specified image.
Members		
<b>PUT</b>	/v1/{tenant_id}/images/{image_id}/members	Replaces a membership list for an image.
<b>PUT</b>	/v1/{tenant_id}/images/{image_id}/members/{owner_id}	Adds a member to an image.
<b>DELETE</b>	/v1/{tenant_id}/images/{image_id}/members/{owner_id}	Removes a member from an image.
Shared images		
<b>GET</b>	/v1/{tenant_id}/shared-images/{owner}	Lists the VM images shared with a specified owner. The owner ID is the tenant ID.

## 13.1. API versions

Method	URI	Description
<b>GET</b>	/	Lists information about all Image Service API versions.

## 13.1.1. List API versions

Method	URI	Description
GET	/	Lists information about all Image Service 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": "v2.2",
            "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": "CURRENT",
            "id": "v1.1",
            "links": [
                {
                    "href": "http://23.253.228.211:9292/v1/",
                    "rel": "self"
                }
            ]
        }
    ]
}
```

```
        },
        {
            "status": "SUPPORTED",
            "id": "v1.0",
            "links": [
                {
                    "href": "http://23.253.228.211:9292/v1/",
                    "rel": "self"
                }
            ]
        }
    ]
```

## 13.2. Images

Method	URI	Description
<b>POST</b>	/v1/{tenant_id}/images	Registers a virtual machine (VM) image.
<b>GET</b>	/v1/{tenant_id}/images/detail{?name,container_format,disk_format,status,size_min,size_max,changes-since}	Lists details for available images.
<b>GET</b>	/v1/{tenant_id}/images/{image_id}	Shows the image details as headers and the image binary in the body of the response.
<b>PUT</b>	/v1/{tenant_id}/images/{image_id}	Updates an image, uploads an image file, or updates metadata for an image.
<b>DELETE</b>	/v1/{tenant_id}/images/{image_id}	Deletes the specified image.

## 13.2.1. Create image

Method	URI	Description
POST	/v1/{tenant_id}/images	Registers a virtual machine (VM) image.

**Normal response codes:** 202

### 13.2.1.1. Request

This table shows the URI parameters for the create image 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.

### 13.2.1.2. Response

This operation does not return a response body.

## 13.2.2. List image details

Method	URI	Description
GET	/v1/{tenant_id}/images/detail{?name,container_format,disk_format,status,size_min,size_max,changes-since}	Lists details for available images.

Normal response codes: 200

### 13.2.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

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

Name	Type	Description
name	String <i>(Optional)</i>	Name of the image as a string.
container_format	ImageStatus <i>(Optional)</i>	Value of the container format, such as ovf, bare, aki, ari, or ami.
disk_format	UUID <i>(Optional)</i>	Value of the disk format.
status	Int <i>(Optional)</i>	Filters the list of images to those with this status. Valid values are queued, saving, active, killed, deleted, and pending_delete.
size_min	String <i>(Optional)</i>	The minimum size of the image, in bytes.
size_max	String <i>(Optional)</i>	Value of the maximum size of the image in bytes.
changes-since	DateTime <i>(Optional)</i>	Filters the list of images to those that have changed since this time stamp value.

This operation does not accept a request body.

### 13.2.2.2. Response

This operation does not return a response body.

## 13.2.3. Show image details and image binary

Method	URI	Description
GET	/v1/{tenant_id}/images/{image_id}	Shows the image details as headers and the image binary in the body of the response.

**Normal response codes:** 200

### 13.2.3.1. Request

This table shows the URI parameters for the show image details and image binary request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{image_id}	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

### 13.2.3.2. Response

#### Example 13.2. Show image details and image binary: JSON response

```
{
    "images": [
        {
            "uri": "http://glance.example.com/images/71c675ab-d94f-49cd-a114-e12490b328d9",
            "name": "Ubuntu 10.04 Plain 5GB",
            "disk_format": "vhd",
            "container_format": "ovf",
            "size": "5368709120",
            "checksum": "c2e5db72bd7fd153f53ede5da5a06de3",
            "created_at": "2010-02-03 09:34:01",
            "updated_at": "2010-02-03 09:34:01",
            "deleted_at": "",
            "status": "active",
            "is_public": true,
            "min_ram": 256,
            "min_disk": 5,
            "owner": null,
            "properties": {
                "distro": "Ubuntu 10.04 LTS"
            }
        },
        {
            "...": ...
        }
    ]
}
```

## 13.2.4. Update image

Method	URI	Description
PUT	/v1/{tenant_id}/images/{image_id}	Updates an image, uploads an image file, or updates metadata for an image.

**Normal response codes:** 200

### 13.2.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{image_id}	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

### 13.2.4.2. Response

This operation does not return a response body.

## 13.2.5. Delete image

Method	URI	Description
<b>DELETE</b>	/v1/{tenant_id}/images/{image_id}	Deletes the specified image.

**Normal response codes:** 204

### 13.2.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{image_id}	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

## 13.3. Members

Method	URI	Description
<b>PUT</b>	/v1/{tenant_id}/images/{image_id}/members	Replaces a membership list for an image.
<b>PUT</b>	/v1/{tenant_id}/images/{image_id}/members/{owner_id}	Adds a member to an image.
<b>DELETE</b>	/v1/{tenant_id}/images/{image_id}/members/{owner_id}	Removes a member from an image.

## 13.3.1. Replace member

Method	URI	Description
PUT	/v1/{tenant_id}/images/{image_id}/members	Replaces a membership list for an image.

**Normal response codes:** 204

### 13.3.1.1. Request

This table shows the URI parameters for the replace member request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{image_id}	String	Image ID stored through the image API. Typically a UUID.

### 13.3.2. Add member

Method	URI	Description
PUT	/v1/{tenant_id}/images/{image_id}/members/{owner_id}	Adds a member to an image.

If you omit the request body from this call, this request adds the membership to the image, leaves the existing memberships unmodified, and sets the `can_share` attribute to false for new memberships.

**Normal response codes:** 204

#### 13.3.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{image_id}	String	Image ID stored through the image API. Typically a UUID.
{owner_id}	String	Owner ID, which is the tenant ID.

#### Example 13.3. Add member: JSON request

```
{
  "members": [
    {
      "member_id": "tenant1",
      "can_share": false
    },
    {
      "member_id": "tenant2",
      "can_share": false
    }
  ]
}
```

### 13.3.3. Delete image

Method	URI	Description
<b>DELETE</b>	/v1/{tenant_id}/images/{image_id}/members/{owner_id}	Removes a member from an image.

**Normal response codes:** 204

#### 13.3.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{image_id}	String	Image ID stored through the image API. Typically a UUID.
{owner_id}	String	Owner ID, which is the tenant ID.

This operation does not accept a request body.

## 13.4. Shared images

Method	URI	Description
<b>GET</b>	/v1/{tenant_id}/shared-images/{owner}	Lists the VM images shared with a specified owner. The owner ID is the tenant ID.

## 13.4.1. List shared images

Method	URI	Description
GET	/v1/{tenant_id}/shared-images/{owner}	Lists the VM images shared with a specified owner. The owner ID is the tenant ID.

Normal response codes: 200

### 13.4.1.1. Request

This table shows the URI parameters for the list shared images request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.

This operation does not accept a request body.

### 13.4.1.2. Response

#### Example 13.4. List shared images: JSON response

```
{
  "shared_images": [
    {
      "image_id": "71c675ab-d94f-49cd-a114-e12490b328d9",
      "can_share": false
    },
    {
      "...": ...
    }
  ]
}
```

# 14. Networking API v2.0 (CURRENT)

Use virtual networking services among devices that 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. 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.

## 14.1. API versions

List information for all Networking API versions and show details about API v2.

Method	URI	Description
GET	/	Lists information about all Networking API versions.
GET	/v2.0	Shows details for Networking API v2.0.

## 14.1.1. List API versions

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

**Normal response codes:** 200, 300

### 14.1.1.1. Request

This operation does not accept a request body.

### 14.1.1.2. Response

#### Example 14.1. List API versions: JSON response

```
{
    "versions": [
        {
            "status": "CURRENT",
            "id": "v2.0",
            "links": [
                {
                    "href": "http://23.253.228.211:9696/v2.0",
                    "rel": "self"
                }
            ]
        }
    ]
}
```

#### Example 14.2. 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.

## 14.1.2. Show API v2.0 details

Method	URI	Description
GET	/v2.0	Shows details for Networking API v2.0.

**Normal response codes:** 200, 203

### 14.1.2.1. Request

This operation does not accept a request body.

### 14.1.2.2. Response

#### Example 14.3. 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 14.4. 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>
```

This operation does not return a response body.

## 14.2. Networks

List, show information for, create, update, and delete networks.

Method	URI	Description
<b>GET</b>	/	Lists information about all Networking API versions.
<b>GET</b>	/v2.0	Shows details for Networking API v2.0.
<b>GET</b>	/v2.0/extensions	Lists available Networking API extensions.
<b>GET</b>	/v2.0/extensions/{alias}	Gets detailed information for a specified extension.
<b>GET</b>	/v2.0/networks	Lists networks to which the specified tenant has access.
<b>POST</b>	/v2.0/networks	Creates a network.
<b>POST</b>	/v2.0/networks	Creates multiple networks in a single request.
<b>GET</b>	/v2.0/networks/{network_id}	Shows information for a specified network.
<b>PUT</b>	/v2.0/networks/{network_id}	Updates a specified network.
<b>DELETE</b>	/v2.0/networks/{network_id}	Deletes a specified network and its associated resources.



## 14.2.1. List API versions

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

**Normal response codes:** 200, 300

### 14.2.1.1. Request

This operation does not accept a request body.

### 14.2.1.2. Response

#### Example 14.5. List API versions: JSON response

```
{
    "versions": [
        {
            "status": "CURRENT",
            "id": "v2.0",
            "links": [
                {
                    "href": "http://23.253.228.211:9696/v2.0",
                    "rel": "self"
                }
            ]
        }
    ]
}
```

#### Example 14.6. 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.

## 14.2.2. Show API v2.0 details

Method	URI	Description
GET	/v2.0	Shows details for Networking API v2.0.

**Normal response codes:** 200, 203

### 14.2.2.1. Request

This operation does not accept a request body.

### 14.2.2.2. Response

#### Example 14.7. 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 14.8. 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.

## 14.2.3. List extensions

Method	URI	Description
GET	/v2.0/extensions	Lists available Networking API extensions.

**Normal response codes:** 200, 203

**Error response codes:** computeFault (400, 500, ...)

### 14.2.3.1. Request

This operation does not accept a request body.

### 14.2.3.2. Response

#### Example 14.9. List extensions: JSON response

```
{
    "extensions": [
        {
            "updated": "2013-01-20T00:00:00-00:00",
            "name": "Neutron Service Type Management",
            "links": [],
            "namespace": "http://docs.openstack.org/ext/neutron/service-type/
api/v1.0",
            "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": [],
            "namespace": "http://docs.openstack.org/ext/securitygroups/api/v2.
0",
            "alias": "security-group",
            "description": "The security groups extension."
        },
        {
            "updated": "2013-02-07T10:00:00-00:00",
            "name": "L3 Agent Scheduler",
            "links": [],
            "namespace": "http://docs.openstack.org/ext/l3_agent_scheduler/
api/v1.0",
            "alias": "l3_agent_scheduler",
            "description": "Schedule routers among l3 agents"
        },
        {
            "updated": "2013-02-07T10:00:00-00:00",
            "name": "Loadbalancer Agent Scheduler",
            "links": [],
            "namespace": "http://docs.openstack.org/ext/lbaas_agent_scheduler/
api/v1.0",
            "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": [],
        "namespace": "http://docs.openstack.org/ext/neutron/ext-gw-mode/
api/v1.0",
        "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": [],
        "namespace": "http://docs.openstack.org/ext/binding/api/v1.0",
        "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": [],
        "namespace": "http://docs.openstack.org/ext/provider/api/v1.0",
        "alias": "provider",
        "description": "Expose mapping of virtual networks to physical
networks"
    },
    {
        "updated": "2013-02-03T10:00:00-00:00",
        "name": "agent",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/agent/api/v2.0",
        "alias": "agent",
        "description": "The agent management extension."
    },
    {
        "updated": "2012-07-29T10:00:00-00:00",
        "name": "Quota management support",
        "links": [],
        "namespace": "http://docs.openstack.org/network/ext/quotas-sets/
api/v2.0",
        "alias": "quotas",
        "description": "Expose functions for quotas management per tenant"
    },
    {
        "updated": "2013-02-07T10:00:00-00:00",
        "name": "DHCP Agent Scheduler",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/dhcp_agent_scheduler/
api/v1.0",
        "alias": "dhcp_agent_scheduler",
        "description": "Schedule networks among dhcp agents"
    },
    {
        "updated": "2013-06-27T10:00:00-00:00",
        "name": "Multi Provider Network",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/multi-provider/api/v1.
0",
        "alias": "multi-provider",
        "description": "Expose multi-provider network abstraction"
    }
]
```

```
        "description": "Expose mapping of virtual networks to multiple
physical networks"
    },
    {
        "updated": "2013-01-14T10:00:00-00:00",
        "name": "Neutron external network",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/neutron/external_net/
api/v1.0",
        "alias": "external-net",
        "description": "Adds external network attribute to network
resource."
    },
    {
        "updated": "2012-07-20T10:00:00-00:00",
        "name": "Neutron L3 Router",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/neutron/router/api/v1.
0",
        "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": [],
        "namespace": "http://docs.openstack.org/ext/allowedaddresspairs/
api/v2.0",
        "alias": "allowed-address-pairs",
        "description": "Provides allowed address pairs"
    },
    {
        "updated": "2013-03-17T12:00:00-00:00",
        "name": "Neutron Extra DHCP opts",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/neutron/
extra_dhcp_opt/api/v1.0",
        "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": [],
        "namespace": "http://wiki.openstack.org/neutron/LBaas/API_1.0",
        "alias": "lbaas",
        "description": "Extension for LoadBalancing service"
    },
    {
        "updated": "2013-02-01T10:00:00-00:00",
        "name": "Neutron Extra Route",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/neutron/extraroutes/
api/v1.0",
        "alias": "extraroute",
        "description": "Extra routes configuration for L3 router"
    }
}
```

```
    ]  
}
```

### Example 14.10. 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"/>  
    <namespace>http://docs.openstack.org/ext/neutron/service-type/api/v1.  
0</namespace>  
    <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"/>  
    <namespace>http://docs.openstack.org/ext/securitygroups/api/v2.0</  
namespace>  
    <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"/>  
    <namespace>http://docs.openstack.org/ext/l3_agent_scheduler/api/v1.0</  
namespace>  
    <alias>l3_agent_scheduler</alias>  
    <description>Schedule routers among l3 agents</description>  
  </extension>  
  <extension>  
    <updated>2013-02-07T10:00:00-00:00</updated>  
    <name>Loadbalancer Agent Scheduler</name>  
    <links quantum:type="list"/>  
    <namespace>http://docs.openstack.org/ext/lbaas_agent_scheduler/api/v1.  
0</namespace>  
    <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"/>  
    <namespace>http://docs.openstack.org/ext/neutron/ext-gw-mode/api/v1.  
0</namespace>  
    <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"/>
<namespace>http://docs.openstack.org/ext/binding/api/v1.0</namespace>
<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"/>
    <namespace>http://docs.openstack.org/ext/provider/api/v1.0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/agent/api/v2.0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/network/ext/quotas-sets/api/v2.
0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/dhcp_agent_scheduler/api/v1.
0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/multi-provider/api/v1.0</
namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/neutron/external_net/api/v1.
0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/neutron/router/api/v1.0</
namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/allowedaddresspairs/api/v2.
0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/neutron/extr dhcp_opt/api/
v1.0</namespace>
    <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"/>
    <namespace>http://wiki.openstack.org/neutron/LBaaS/API_1.0</namespace>
    <alias>lbaas</alias>
    <description>Extension for LoadBalancing services</description>
</extension>
<extension>
    <updated>2013-02-01T10:00:00-00:00</updated>
    <name>Neutron Extra Route</name>
    <links quantum:type="list"/>
    <namespace>http://docs.openstack.org/ext/neutron/extraroutes/api/v1.
0</namespace>
    <alias>extraroute</alias>
    <description>Extra routes configuration for L3
        router</description>
</extension>
</extensions>
```

This operation does not return a response body.

## 14.2.4. Get extension details

Method	URI	Description
GET	/v2.0/extensions/{alias}	Gets detailed information for a specified extension.

**Normal response codes:** 200, 203

**Error response codes:** computeFault (400, 500, ...)

### 14.2.4.1. Request

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

Name	Type	Description
{alias}	String	

This operation does not accept a request body.

### 14.2.4.2. Response

#### Example 14.11. Get extension details: JSON response

```
{
  "extension": {
    "updated": "2013-02-03T10:00:00-00:00",
    "name": "agent",
    "links": [],
    "namespace": "http://docs.openstack.org/ext/agent/api/v2.0",
    "alias": "agent",
    "description": "The agent management extension."
  }
}
```

#### Example 14.12. 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"/>
  <namespace>http://docs.openstack.org/ext/agent/api/v2.0</namespace>
  <alias>agent</alias>
  <description>The agent management extension.</description>
</extension>
```

This operation does not return a response body.

## 14.2.5. List networks

Method	URI	Description
GET	/v2.0/networks	Lists networks to which the specified tenant has access.

You can control which attributes are returned by using the fields query parameter. For information, see [Filtering and Column Selection](#) in the *OpenStack Networking API v2.0 Reference*.

**Normal response codes:** 200

**Error response codes:** unauthorized (401)

### 14.2.5.1. Request

This operation does not accept a request body.

### 14.2.5.2. Response

#### Example 14.13. 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 14.14. 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>
```

## 14.2.6. Create network

Method	URI	Description
POST	/v2.0/networks	Creates a network.

This operation does not require a request body. The tenant ID that you specify in the URI is the tenant that creates the network. An admin user can specify another tenant ID in the optional request body, which is the tenant who owns the network.

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401)

### 14.2.6.1. 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>
```

### 14.2.6.2. Response

#### Example 14.17. Create network: JSON response

```
{
    "network": {
        "status": "ACTIVE",
        "subnets": [],
        "name": "net1",
        "admin_state_up": true,
        "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
        "segments": [
            {
                "provider:segmentation_id": 2,
                "provider:physical_network": "8bab8453-1bc9-45af-8c70-
f83aa9b50453",
                "provider:network_type": "vlan"
            },
            {
                "provider:segmentation_id": null,
                "provider:physical_network": "8bab8453-1bc9-45af-8c70-
f83aa9b50453",
                "provider:network_type": "stt"
            }
        ]
    }
}
```

```
        ],
        "shared": false,
        "port_security_enabled": true,
        "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>
```

## 14.2.7. Bulk create networks

Method	URI	Description
POST	/v2.0/networks	Creates multiple networks in a single request.

In the request body, specify a list of networks.

The bulk create operation is always atomic. Either all or no networks in the request body are created.

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401)

### 14.2.7.1. Request

#### Example 14.19. Bulk create networks: JSON request

```
{
  "networks": [
    {
      "name": "sample_network3",
      "admin_state_up": true
    },
    {
      "name": "sample_network4",
      "admin_state_up": true
    }
  ]
}
```

#### Example 14.20. 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>
```

### 14.2.7.2. Response

#### Example 14.21. Bulk create networks: JSON response

```
{
  "networks": [
    {
      "status": "ACTIVE",
      "subnets": [],
      "name": "sample_network3",
      "provider:physical_network": null,
      "admin_state_up": true,
      "id": "a2345678-1234-5678-9012-345678901234"
    }
  ]
}
```

```

        "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-4ald-b799-ab73111476e2",
        "provider:segmentation_id": null
    }
]
}

```

### Example 14.22. 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>

```

## 14.2.8. Show network

Method	URI	Description
GET	/v2.0/networks/{network_id}	Shows information for a specified network.

You can control which attributes are returned by using the fields query parameter. For information, see [Filtering and Column Selection](#) in the *OpenStack Networking API v2.0 Reference*.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), itemNotFound (404)

### 14.2.8.1. Request

This table shows the URI parameters for the show 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.

### 14.2.8.2. Response

#### Example 14.23. Show network: JSON response

```
{
  "network": {
    "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
  }
}
```

#### Example 14.24. Show 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>
```

```
<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>
<router:external quantum:type="bool">True</router:external>
<shared quantum:type="bool">True</shared>
<id>d32019d3-bc6e-4319-9c1d-6722fc136a22</id>
<provider:segmentation_id xsi:nil="true"/>
</network>
```

## 14.2.9. Update network

Method	URI	Description
PUT	/v2.0/networks/{network_id}	Updates a specified network.

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

### 14.2.9.1. 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 14.25. Update network: JSON request

```
{
  "network": {
    "name": "sample_network_5_updated"
  }
}
```

#### Example 14.26. 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>
```

### 14.2.9.2. Response

#### Example 14.27. 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,
    "shared": false,
    "id": "1f370095-98f6-4079-be64-6d3d4a6adcc6",
    "provider:segmentation_id": null
  }
}
```

{

**Example 14.28. 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>
```

## 14.2.10. Delete network

Method	URI	Description
<b>DELETE</b>	/v2.0/networks/{network_id}	Deletes a specified network and its associated resources.

**Normal response codes:** 204

**Error response codes:** unauthorized (401), itemNotFound (404), conflict (409)

### 14.2.10.1. 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.

## 14.3. Subnets

List, show information for, create, update, and delete subnet resources.

Method	URI	Description
<b>GET</b>	/v2.0/subnets	Lists subnets to which the specified tenant has access.
<b>POST</b>	/v2.0/subnets	Creates a subnet on a specified network.
<b>POST</b>	/v2.0/subnets	Creates multiple subnets in a single request. Specify a list of subnets in the request body.
<b>GET</b>	/v2.0/subnets/{subnet_id}	Shows information for a specified subnet.
<b>PUT</b>	/v2.0/subnets/{subnet_id}	Updates a specified subnet.
<b>DELETE</b>	/v2.0/subnets/{subnet_id}	Deletes a specified subnet.

## 14.3.1. List subnets

Method	URI	Description
GET	/v2.0/subnets	Lists subnets to which the specified tenant has access.

Default policy settings returns exclusively subnets owned by the tenant submitting the request, unless the request is submitted by an 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. For information, see [Filtering and Column Selection](#) in the *OpenStack Networking API v2.0 Reference*.

**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 14.29. 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": []
        }
    ]
}
```

```
        "host_routes": [],
        "ip_version": 4,
        "gateway_ip": "192.0.0.1",
        "cidr": "192.0.0.0/8",
        "id": "54d6f61d-db07-451c-9ab3-b9609b6b6f0b"
    }
]
}
```

### Example 14.30. List subnets: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<subnets 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">
    <subnet>
        <name>private-subnet</name>
        <enable_dhcp quantum:type="bool">True</enable_dhcp>
        <network_id>db193ab3-96e3-4cb3-8fc5-05f4296d0324</network_id>
        <tenant_id>26a7980765d0414dbc1fc1f88cdb7e6e</tenant_id>
        <dns_nameservers quantum:type="list"/>
        <allocation_pools>
            <allocation_pool>
                <start>10.0.0.2</start>
                <end>10.0.0.254</end>
            </allocation_pool>
        </allocation_pools>
        <host_routes quantum:type="list"/>
        <ip_version quantum:type="long">4</ip_version>
        <gateway_ip>10.0.0.1</gateway_ip>
        <cidr>10.0.0.0/24</cidr>
        <id>08eae331-0402-425a-923c-34f7cfe39c1b</id>
    </subnet>
    <subnet>
        <name>my_subnet</name>
        <enable_dhcp quantum:type="bool">True</enable_dhcp>
        <network_id>d32019d3-bc6e-4319-9c1d-6722fc136a22</network_id>
        <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
        <dns_nameservers quantum:type="list"/>
        <allocation_pools>
            <allocation_pool>
                <start>192.0.0.2</start>
                <end>192.255.255.254</end>
            </allocation_pool>
        </allocation_pools>
        <host_routes quantum:type="list"/>
        <ip_version quantum:type="long">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.

## 14.3.2. Create subnet

Method	URI	Description
POST	/v2.0/subnets	Creates a subnet on a specified network.

By default, OpenStack Networking creates IP v4 subnets. To create an IP v6 subnet, you must specify the value 6 for the `ip_version` attribute in the request body. OpenStack Networking does not try to derive the correct IP version from the provided CIDR. If the parameter for the gateway address, `gateway_ip`, is not specified, OpenStack Networking allocates an address from the cidr for the gateway for the subnet.

To specify a subnet without a gateway, specify the value `null` for the `gateway_ip` attribute in the request body. If allocation pools attribute, `allocation_pools`, is not specified, 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 `allocation_pools` and `gateway_ip` are both specified, it is up to the user to ensure that the gateway IP does not overlap with the specified allocation pools; otherwise a 409 Conflict error occurs.

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), conflict (409)

### 14.3.2.1. Request

#### Example 14.31. Create subnet: JSON request

```
{
    "subnet": {
        "network_id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
        "ip_version": 4,
        "cidr": "192.168.199.0/24"
    }
}
```

#### Example 14.32. Create subnet: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<subnet>
    <name>test_subnet_1</name>
    <network_id>d32019d3-bc6e-4319-9c1d-6722fc136a22</network_id>
    <cidr>192.0.0.0/8</cidr>
    <ip_version>4</ip_version>
</subnet>
```

This operation does not accept a request body.

### 14.3.2.2. Response

#### Example 14.33. 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 14.34. Create subnet: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<subnet 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">
    <name>test_subnet_1</name>
    <enable_dhcp quantum:type="bool">True</enable_dhcp>
    <network_id>d32019d3-bc6e-4319-9c1d-6722fc136a22</network_id>
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
    <dns_nameservers quantum:type="list"/>
    <allocation_pools>
        <allocation_pool>
            <start>192.0.0.2</start>
            <end>192.255.255.254</end>
        </allocation_pool>
    </allocation_pools>
    <host_routes quantum:type="list"/>
    <ip_version quantum:type="int">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.

### 14.3.3. Bulk create subnet

Method	URI	Description
POST	/v2.0/subnets	Creates multiple subnets in a single request. Specify a list of subnets in the request body.

The bulk create operation is always atomic. Either all or no subnets in the request body are created.

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), conflict (409)

#### 14.3.3.1. Request

##### Example 14.35. 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 14.36. 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.

### 14.3.3.2. Response

#### Example 14.37. 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 14.38. Bulk create subnet: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<subnets 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">
    <subnet>
        <name>test_subnet_1</name>
        <enable_dhcp quantum:type="bool">True</enable_dhcp>
        <network_id>a3775a7d-9f8b-4148-be81-c84bbd0837ce</network_id>
        <tenant_id>60cd4f6dbc2f491982a284e7b83b5be3</tenant_id>
        <dns_nameservers quantum:type="list"/>
        <allocation_pools>
            <allocation_pool>
```

```
        <start>10.0.0.2</start>
        <end>10.255.255.254</end>
    </allocation_pool>
</allocation_pools>
<host_routes quantum:type="list"/>
<ip_version quantum:type="int">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 quantum:type="bool">True</enable_dhcp>
    <network_id>a3775a7d-9f8b-4148-be81-c84bbd0837ce</network_id>
    <tenant_id>60cd4f6dbc2f491982a284e7b83b5be3</tenant_id>
    <dns_nameservers quantum:type="list"/>
    <allocation_pools>
        <allocation_pool>
            <start>192.168.0.2</start>
            <end>192.168.255.254</end>
        </allocation_pool>
    </allocation_pools>
    <host_routes quantum:type="list"/>
    <ip_version quantum:type="int">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.

## 14.3.4. Show subnet

Method	URI	Description
GET	/v2.0/subnets/{subnet_id}	Shows information for a specified subnet.

You can control which attributes are returned by using the fields query parameter. For information, see [Filtering and Column Selection](#) in the *OpenStack Networking API v2.0 Reference*.

**Normal response codes:** 201

**Error response codes:** unauthorized (401), itemNotFound (404)

### 14.3.4.1. Request

This table shows the URI parameters for the show 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.

### 14.3.4.2. Response

#### Example 14.39. Show subnet: 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 14.40. Show subnet: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<subnet 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">
    <name>test_subnet_1</name>
```

```
<enable_dhcp quantum:type="bool">True</enable_dhcp>
<network_id>d32019d3-bc6e-4319-9c1d-6722fc136a22</network_id>
<tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
<dns_nameservers quantum:type="list"/>
<allocation_pools>
    <allocation_pool>
        <start>192.0.0.2</start>
        <end>192.255.255.254</end>
    </allocation_pool>
</allocation_pools>
<host_routes quantum:type="list"/>
<ip_version quantum:type="long">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.

## 14.3.5. Update subnet

Method	URI	Description
PUT	/v2.0/subnets/{subnet_id}	Updates a specified 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:** 201

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

### 14.3.5.1. Request

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

Name	Type	Description
{subnet_id}	UUID	The UUID for the subnet of interest to you.

#### Example 14.41. Update subnet: JSON request

```
{
    "subnet": {
        "name": "my_subnet"
    }
}
```

#### Example 14.42. 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.

### 14.3.5.2. Response

#### Example 14.43. Update subnet: JSON response

```
{
    "subnet": {
        "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"
            }
        ],
        "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 14.44. Update subnet: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<subnet 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">
    <name>my_subnet</name>
    <enable_dhcp quantum:type="bool">True</enable_dhcp>
    <network_id>d32019d3-bc6e-4319-9c1d-6722fc136a22</network_id>
    <tenant_id>4fd44f30292945e481c7b8a0c8908869</tenant_id>
    <dns_nameservers quantum:type="list"/>
    <allocation_pools>
        <allocation_pool>
            <start>192.0.0.2</start>
            <end>192.255.255.254</end>
        </allocation_pool>
    </allocation_pools>
    <host_routes quantum:type="list"/>
    <ip_version quantum:type="long">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.

## 14.3.6. Delete subnet

Method	URI	Description
<b>DELETE</b>	/v2.0/subnets/{subnet_id}	Deletes a specified subnet.

The operation fails if subnet IP addresses are still allocated.

**Normal response codes:** 204

**Error response codes:** unauthorized (401), itemNotFound (404), conflict (409)

### 14.3.6.1. Request

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

Name	Type	Description
{subnet_id}	UUID	The UUID for the subnet of interest to you.

This operation does not accept a request body.

## 14.4. Ports

List, show information for, create, update, and delete ports.

Method	URI	Description
<b>GET</b>	/v2.0/ports	Lists ports to which the tenant has access.
<b>POST</b>	/v2.0/ports	Creates a port on a specified network.
<b>POST</b>	/v2.0/ports	Creates multiple ports in a single request. Specify a list of ports in the request body.
<b>GET</b>	/v2.0/ports/{port_id}	Shows information for a specified port.
<b>PUT</b>	/v2.0/ports/{port_id}	Updates a specified port.
<b>DELETE</b>	/v2.0/ports/{port_id}	Deletes a specified port.

## 14.4.1. List ports

Method	URI	Description
GET	/v2.0/ports	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 an 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](#) in the *OpenStack Networking API v2.0 Reference*.

**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 14.45. 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",
            "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",
        "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-a570ede1d159",
        "security_groups": [],
        "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824"
    }
]
}

```

### Example 14.46. 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>
    </port>

```

```
</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-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.

## 14.4.2. Create port

Method	URI	Description
POST	/v2.0/ports	Creates a port on a specified network.

You must specify the network where the port is to created in the `network_id` attribute in the request body.

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), macGenerationFailure (503), serviceUnavailable (503)

### 14.4.2.1. Request

#### Example 14.47. Create port: JSON request

```
{
  "port": {
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "name": "private-port",
    "admin_state_up": true
  }
}
```

#### Example 14.48. 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.

### 14.4.2.2. Response

#### Example 14.49. 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": "20a1a2a3-a4b5-4cde-8f9b-1234567890ab",
        "ip_address": "192.168.1.100"
      }
    ]
  }
}
```

```
{  
    "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 14.50. 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.

## 14.4.3. Bulk create ports

Method	URI	Description
POST	/v2.0/ports	Creates multiple ports in a single request. Specify a list of ports in the request body.

Guarantees the atomic completion of the bulk operation.

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), conflict (409), macGenerationFailure (503)

### 14.4.3.1. Request

#### Example 14.51. 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 14.52. 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.

### 14.4.3.2. Response

#### Example 14.53. Bulk create ports: JSON response

```
{
  "ports": [
    {
```

```

    "status": "DOWN",
    "binding:host_id": "",
    "name": "sample_port_1",
    "allowed_address_pairs": [],
    "admin_state_up": false,
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "tenant_id": "d6700c0c9ffa4f1cb322cd4alf3906fa",
    "binding:vif_details": {},
    "binding:vnic_type": "normal",
    "binding:vif_type": "unbound",
    "device_owner": "",
    "mac_address": "fa:16:3e:48:b8:9f",
    "binding:profile": {},
    "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",
    "binding:host_id": "",
    "name": "sample_port_2",
    "allowed_address_pairs": [],
    "admin_state_up": false,
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "tenant_id": "d6700c0c9ffa4f1cb322cd4alf3906fa",
    "binding:vif_details": {},
    "binding:vnic_type": "normal",
    "binding:vif_type": "unbound",
    "device_owner": "",
    "mac_address": "fa:16:3e:f4:73:df",
    "binding:profile": {},
    "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 14.54. Bulk create 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>DOWN</status>
    <binding:host_id/>
    <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>
    <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:fa:e2:34</mac_address>
    <binding:profile quantum:type="dict"/>
    <fixed_ips>
        <fixed_ip>
            <subnet_id>a0304c3a-4f08-4c43-88af-d796509c97d2</subnet_id>
            <ip_address>10.0.0.7</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>
    <binding:host_id/>
    <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>
    <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:e6:cf:d9</mac_address>
    <binding:profile quantum:type="dict"/>
    <fixed_ips>
        <fixed_ip>
            <subnet_id>a0304c3a-4f08-4c43-88af-d796509c97d2</subnet_id>
            <ip_address>10.0.0.8</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.

## 14.4.4. Show port

Method	URI	Description
GET	/v2.0/ports/{port_id}	Shows information for a specified port.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), itemNotFound (404)

### 14.4.4.1. Request

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

Name	Type	Description
{port_id}	UUID	The UUID for the port of interest to you.

This operation does not accept a request body.

### 14.4.4.2. Response

#### Example 14.55. Show port: 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",
    "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 14.56. Show 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>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.

## 14.4.5. Update port

Method	URI	Description
PUT	/v2.0/ports/{port_id}	Updates a specified 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 subnets allocation pools, and replaced by the IPs specified in the body for the update request. Therefore, this operation replaces the `fixed_ip` attribute when it is specified 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 specified in the body for the update request. Therefore, this operation replaces the `security_groups` attribute when you specify it in the request body. However, if the specified security groups are not valid, the operation fails and the existing security groups are not removed from the port.

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404), conflict (409)

### 14.4.5.1. Request

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

Name	Type	Description
{port_id}	UUID	The UUID for the port of interest to you.

#### Example 14.57. Update port: JSON request

```
{
  "port": {
    "network_id": "a87cc70a-3e15-4acf-8205-9b711a3531b7",
    "name": "private-port",
    "admin_state_up": true
  }
}
```

#### Example 14.58. Update port: JSON 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.

### 14.4.5.2. Response

#### Example 14.59. 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 14.60. 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.

## 14.4.6. Delete port

Method	URI	Description
<b>DELETE</b>	/v2.0/ports/{port_id}	Deletes a specified port.

Any IP addresses that are associated with the port are returned to the respective subnets allocation pools.

**Normal response codes:** 204

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 14.4.6.1. Request

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

Name	Type	Description
{port_id}	UUID	The UUID for the port of interest to you.

This operation does not accept a request body.

# 15. Networking API v2.0 extensions

## 15.1. Extensions

List available Networking API v2.0 extensions and show details for a specified extension.

Method	URI	Description
<b>GET</b>	/v2.0/extensions	Lists available Networking API extensions.
<b>GET</b>	/v2.0/extensions/{alias}	Gets detailed information for a specified extension.

## 15.1.1. List extensions

Method	URI	Description
GET	/v2.0/extensions	Lists available Networking API extensions.

**Normal response codes:** 200, 203

**Error response codes:** computeFault (400, 500, ...)

### 15.1.1.1. Request

This operation does not accept a request body.

### 15.1.1.2. Response

#### Example 15.1. List extensions: JSON response

```
{
    "extensions": [
        {
            "updated": "2013-01-20T00:00:00-00:00",
            "name": "Neutron Service Type Management",
            "links": [],
            "namespace": "http://docs.openstack.org/ext/neutron/service-type/
api/v1.0",
            "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": [],
            "namespace": "http://docs.openstack.org/ext/securitygroups/api/v2.
0",
            "alias": "security-group",
            "description": "The security groups extension."
        },
        {
            "updated": "2013-02-07T10:00:00-00:00",
            "name": "L3 Agent Scheduler",
            "links": [],
            "namespace": "http://docs.openstack.org/ext/l3_agent_scheduler/
api/v1.0",
            "alias": "l3_agent_scheduler",
            "description": "Schedule routers among l3 agents"
        },
        {
            "updated": "2013-02-07T10:00:00-00:00",
            "name": "Loadbalancer Agent Scheduler",
            "links": [],
            "namespace": "http://docs.openstack.org/ext/lbaas_agent_scheduler/
api/v1.0",
            "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": [],
        "namespace": "http://docs.openstack.org/ext/neutron/ext-gw-mode/
api/v1.0",
        "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": [],
        "namespace": "http://docs.openstack.org/ext/binding/api/v1.0",
        "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": [],
        "namespace": "http://docs.openstack.org/ext/provider/api/v1.0",
        "alias": "provider",
        "description": "Expose mapping of virtual networks to physical
networks"
    },
    {
        "updated": "2013-02-03T10:00:00-00:00",
        "name": "agent",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/agent/api/v2.0",
        "alias": "agent",
        "description": "The agent management extension."
    },
    {
        "updated": "2012-07-29T10:00:00-00:00",
        "name": "Quota management support",
        "links": [],
        "namespace": "http://docs.openstack.org/network/ext/quotas-sets/
api/v2.0",
        "alias": "quotas",
        "description": "Expose functions for quotas management per tenant"
    },
    {
        "updated": "2013-02-07T10:00:00-00:00",
        "name": "DHCP Agent Scheduler",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/dhcp_agent_scheduler/
api/v1.0",
        "alias": "dhcp_agent_scheduler",
        "description": "Schedule networks among dhcp agents"
    },
    {
        "updated": "2013-06-27T10:00:00-00:00",
        "name": "Multi Provider Network",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/multi-provider/api/v1.
0",
        "alias": "multi-provider",
        "description": "Expose multi-provider network abstraction"
    }
]
```

```
        "description": "Expose mapping of virtual networks to multiple
physical networks"
    },
    {
        "updated": "2013-01-14T10:00:00-00:00",
        "name": "Neutron external network",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/neutron/external_net/
api/v1.0",
        "alias": "external-net",
        "description": "Adds external network attribute to network
resource."
    },
    {
        "updated": "2012-07-20T10:00:00-00:00",
        "name": "Neutron L3 Router",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/neutron/router/api/v1.
0",
        "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": [],
        "namespace": "http://docs.openstack.org/ext/allowedaddresspairs/
api/v2.0",
        "alias": "allowed-address-pairs",
        "description": "Provides allowed address pairs"
    },
    {
        "updated": "2013-03-17T12:00:00-00:00",
        "name": "Neutron Extra DHCP opts",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/neutron/
extra_dhcp_opt/api/v1.0",
        "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": [],
        "namespace": "http://wiki.openstack.org/neutron/LBaas/API_1.0",
        "alias": "lbaas",
        "description": "Extension for LoadBalancing service"
    },
    {
        "updated": "2013-02-01T10:00:00-00:00",
        "name": "Neutron Extra Route",
        "links": [],
        "namespace": "http://docs.openstack.org/ext/neutron/extraroutes/
api/v1.0",
        "alias": "extraroute",
        "description": "Extra routes configuration for L3 router"
    }
}
```

```
    ]  
}
```

### Example 15.2. 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"/>  
    <namespace>http://docs.openstack.org/ext/neutron/service-type/api/v1.  
0</namespace>  
    <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"/>  
    <namespace>http://docs.openstack.org/ext/securitygroups/api/v2.0</  
namespace>  
    <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"/>  
    <namespace>http://docs.openstack.org/ext/l3_agent_scheduler/api/v1.0</  
namespace>  
    <alias>l3_agent_scheduler</alias>  
    <description>Schedule routers among l3 agents</description>  
  </extension>  
  <extension>  
    <updated>2013-02-07T10:00:00-00:00</updated>  
    <name>Loadbalancer Agent Scheduler</name>  
    <links quantum:type="list"/>  
    <namespace>http://docs.openstack.org/ext/lbaas_agent_scheduler/api/v1.  
0</namespace>  
    <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"/>  
    <namespace>http://docs.openstack.org/ext/neutron/ext-gw-mode/api/v1.  
0</namespace>  
    <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"/>
<namespace>http://docs.openstack.org/ext/binding/api/v1.0</namespace>
<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"/>
    <namespace>http://docs.openstack.org/ext/provider/api/v1.0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/agent/api/v2.0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/network/ext/quotas-sets/api/v2.
0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/dhcp_agent_scheduler/api/v1.
0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/multi-provider/api/v1.0</
namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/neutron/external_net/api/v1.
0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/neutron/router/api/v1.0</
namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/allowedaddresspairs/api/v2.
0</namespace>
    <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"/>
    <namespace>http://docs.openstack.org/ext/neutron/extr dhcp_opt/api/
v1.0</namespace>
    <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"/>
    <namespace>http://wiki.openstack.org/neutron/LBaaS/API_1.0</namespace>
    <alias>lbaas</alias>
    <description>Extension for LoadBalancing services</description>
</extension>
<extension>
    <updated>2013-02-01T10:00:00-00:00</updated>
    <name>Neutron Extra Route</name>
    <links quantum:type="list"/>
    <namespace>http://docs.openstack.org/ext/neutron/extraroutes/api/v1.
0</namespace>
    <alias>extraroute</alias>
    <description>Extra routes configuration for L3
        router</description>
</extension>
</extensions>
```

This operation does not return a response body.

## 15.1.2. Get extension details

Method	URI	Description
GET	/v2.0/extensions/{alias}	Gets detailed information for a specified extension.

**Normal response codes:** 200, 203

**Error response codes:** computeFault (400, 500, ...)

### 15.1.2.1. Request

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

Name	Type	Description
{alias}	String	

This operation does not accept a request body.

### 15.1.2.2. Response

#### Example 15.3. Get extension details: JSON response

```
{
  "extension": {
    "updated": "2013-02-03T10:00:00-00:00",
    "name": "agent",
    "links": [],
    "namespace": "http://docs.openstack.org/ext/agent/api/v2.0",
    "alias": "agent",
    "description": "The agent management extension."
  }
}
```

#### Example 15.4. 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"/>
  <namespace>http://docs.openstack.org/ext/agent/api/v2.0</namespace>
  <alias>agent</alias>
  <description>The agent management extension.</description>
</extension>
```

This operation does not return a response body.

## 15.2. Quotas extension (quotas)

List, show information for, update, and reset quotas.

Method	URI	Description
<b>GET</b>	/v2.0/quotas	Lists quotas for tenants who have non-default quota values.
<b>GET</b>	/v2.0/quotas/{tenant}	Shows quotas for a specified tenant.
<b>PUT</b>	/v2.0/quotas/{tenant}	Updates quotas for a specified tenant. Use when non-default quotas are desired.
<b>DELETE</b>	/v2.0/quotas/{tenant}	Resets quotas to default values for a specified tenant.

## 15.2.1. 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)

### 15.2.1.1. Request

This operation does not accept a request body.

### 15.2.1.2. Response

#### Example 15.5. List quotas: JSON response

```
{  
    "quotas": [  
        {  
            "subnet": 10,  
            "network": 10,  
            "floatingip": 50,  
            "tenant_id": "b7445f221cda4f4a8ac7db6b218b1339",  
            "router": 10,  
            "port": 30  
        }  
    ]  
}
```

## 15.2.2. Show quota

Method	URI	Description
GET	/v2.0/quotas	Shows quotas for a specified tenant.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403)

### 15.2.2.1. Request

This operation does not accept a request body.

### 15.2.2.2. Response

#### Example 15.6. Show quota: JSON response

```
{  
    "quota": {  
        "subnet": 10,  
        "router": 10,  
        "port": 50,  
        "network": 10,  
        "floatingip": 50  
    }  
}
```

## 15.2.3. Update quota

Method	URI	Description
PUT	/v2.0/quotas	Updates quotas for a specified tenant. Use when non-default quotas are desired.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403)

### 15.2.3.1. Request

#### Example 15.7. Update quota: JSON request

```
{  
    "quota": {  
        "subnet": 40,  
        "router": 50,  
        "network": 10,  
        "floatingip": 30,  
        "port": 30  
    }  
}
```

### 15.2.3.2. Response

#### Example 15.8. Update quota: JSON response

```
{  
    "quota": {  
        "subnet": 40,  
        "router": 50,  
        "port": 30,  
        "network": 10,  
        "floatingip": 30  
    }  
}
```

## 15.2.4. Reset quota

Method	URI	Description
<b>DELETE</b>	/v2.0/quotas	Resets quotas to default values for a specified tenant.

**Normal response codes:** 204

**Error response codes:** unauthorized (401), forbidden (403)

### 15.2.4.1. Request

This operation does not accept a request body.

## 15.3. Networks provider extended attributes (networks)

Method	URI	Description
<b>GET</b>	/v2.0/networks	Lists networks that are accessible to the tenant who submits the request.
<b>POST</b>	/v2.0/networks	Creates a network.
<b>GET</b>	/v2.0/networks/{network_id}	Shows details for a specified network.
<b>PUT</b>	/v2.0/networks/{network_id}	Updates a specified network.
<b>DELETE</b>	/v2.0/networks/{network_id}	Deletes a specified network.

## 15.3.1. 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

### 15.3.1.1. Request

This operation does not accept a request body.

### 15.3.1.2. Response

#### Example 15.9. List networks: JSON response

```
{
  "network": {
    "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
  }
}
```

#### Example 15.10. 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>
  <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">True</router:external>
  <shared quantum:type="bool">True</shared>
  <id>d32019d3-bc6e-4319-9c1d-6722fc136a22</id>
  <provider:segmentation_id xsi:nil="true"/>
</network>
```

This operation does not return a response body.

## 15.3.2. Create network

Method	URI	Description
POST	/v2.0/networks	Creates a network.

**Normal response codes:** 201

### 15.3.2.1. Request

#### Example 15.11. Create network: JSON request

```
{
    "network": {
        "name": "sample_network",
        "admin_state_up": true
    }
}
```

#### Example 15.12. 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.

### 15.3.2.2. Response

#### Example 15.13. Create network: JSON response

```
{
    "network": {
        "status": "ACTIVE",
        "subnets": [],
        "name": "net1",
        "admin_state_up": true,
        "tenant_id": "9bacb3c5d39d41a79512987f338cf177",
        "segments": [
            {
                "provider:segmentation_id": 2,
                "provider:physical_network": "8bab8453-1bc9-45af-8c70-
f83aa9b50453",
                "provider:network_type": "vlan"
            },
            {
                "provider:segmentation_id": null,
                "provider:physical_network": "8bab8453-1bc9-45af-8c70-
f83aa9b50453",
                "provider:network_type": "stt"
            }
        ],
        "shared": false,
        "port_security_enabled": true,
        "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c"
    }
}
```

```
    }  
}
```

### Example 15.14. 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.

### 15.3.3. Show network details

Method	URI	Description
GET	/v2.0/networks/{network_id}	Shows details for a specified network.

**Normal response codes:** 200

#### 15.3.3.1. 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.

#### 15.3.3.2. Response

##### Example 15.15. Show network details: JSON response

```
{
    "network": {
        "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
    }
}
```

##### Example 15.16. 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>
    <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">True</router:external>
<shared quantum:type="bool">True</shared>
<id>d32019d3-bc6e-4319-9c1d-6722fc136a22</id>
<provider:segmentation_id xsi:nil="true"/>
</network>
```

This operation does not return a response body.

## 15.3.4. Update network

Method	URI	Description
PUT	/v2.0/networks/{network_id}	Updates a specified network.

**Normal response codes:** 201

### 15.3.4.1. 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 15.17. Update network: JSON request

```
{
    "network": {
        "name": "sample_network_5_updated"
    }
}
```

#### Example 15.18. 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.

### 15.3.4.2. Response

#### Example 15.19. 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,
        "shared": false,
        "id": "1f370095-98f6-4079-be64-6d3d4a6adcc6",
        "provider:segmentation_id": null
    }
}
```

### Example 15.20. 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.

## 15.3.5. Delete network

Method	URI	Description
<b>DELETE</b>	/v2.0/networks/{network_id}	Deletes a specified network.

### 15.3.5.1. 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.

## 15.4. 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 specified network, a VXLAN tunnel ID, and a VLAN.

Method	URI	Description
<b>GET</b>	/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.
<b>POST</b>	/v2.0/networks	Creates a network with multiple segment mappings.
<b>GET</b>	/v2.0/networks/{network_id}	Shows details for a specified network with multiple segments.

## 15.4.1. List networks

Method	URI	Description
<b>GET</b>	/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

### 15.4.1.1. Request

This operation does not accept a request body.

### 15.4.1.2. Response

#### Example 15.21. 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,
            "port_security_enabled": true,
            "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c"
        },
        {
            "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,
            "port_security_enabled": true
        }
    ]
}
```

```
        "id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",
        "provider:segmentation_id": null
    }
]
```

## 15.4.2. Create network with multiple segment mappings

Method	URI	Description
POST	/v2.0/networks	Creates a network with multiple segment mappings.

**Normal response codes:** 201

### 15.4.2.1. Request

#### Example 15.22. Create network with multiple segment mappings: JSON request

```
{
  "network": {
    "segments": [
      {
        "provider:segmentation_id": "2",
        "provider:physical_network": "8bab8453-1bc9-45af-8c70-
f83aa9b50453",
        "provider:network_type": "vlan"
      },
      {
        "provider:physical_network": "8bab8453-1bc9-45af-8c70-
f83aa9b50453",
        "provider:network_type": "stt"
      }
    ],
    "name": "net1",
    "admin_state_up": true
  }
}
```

### 15.4.2.2. Response

#### Example 15.23. 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:network_type": "stt"
    }
],
"shared": false,
"port_security_enabled": true,
"id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c"
}
```

## 15.4.3. Show details for a network with multiple segments

Method	URI	Description
GET	/v2.0/networks/{network_id}	Shows details for a specified network with multiple segments.

**Normal response codes:** 200

### 15.4.3.1. 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.

### 15.4.3.2. Response

#### Example 15.24. 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,
        "port_security_enabled": true,
        "id": "4e8e5957-649f-477b-9e5b-f1f75b21c03c"
    }
}
```

## 15.5. Ports binding extended attributes (ports)

Method	URI	Description
GET	/v2.0/ports	Lists ports to which the tenant has access.

Method	URI	Description
<b>POST</b>	/v2.0/ports	Creates a port on the specified network.
<b>GET</b>	/v2.0/ports/{port_id}	Shows information for the specified port.
<b>PUT</b>	/v2.0/ports/{port_id}	Updates the specified port.

## 15.5.1. List ports

Method	URI	Description
GET	/v2.0/ports	Lists ports to which the tenant has access.

**Normal response codes:** 200

### 15.5.1.1. Request

This operation does not accept a request body.

### 15.5.1.2. Response

#### Example 15.25. 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",
      "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
      }
    }
  ]
}
```

```

        "ovs_hybrid_plug": true
    },
    "binding:vif_type": "ovs",
    "device_owner": "network:router_interface",
    "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-a570ede1d159",
    "security_groups": [],
    "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824"
}
]
}

```

### Example 15.26. 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-a570edel1d159</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.

## 15.5.2. Create port

Method	URI	Description
POST	/v2.0/ports	Creates a port on the specified network.

**Normal response codes:** 200

### 15.5.2.1. Request

#### Example 15.27. Create port: JSON request

```
{
    "port": {
        "network_id": "ee2d3158-3e80-4fb3-ba87-c99f515d85e7",
        "admin_state_up": true
    }
}
```

### 15.5.2.2. Response

#### Example 15.28. 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 15.29. 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.

## 15.5.3. Show port

Method	URI	Description
GET	/v2.0/ports/{port_id}	Shows information for the specified port.

**Normal response codes:** 200

### 15.5.3.1. Request

This table shows the URI parameters for the show 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.

### 15.5.3.2. Response

#### Example 15.30. Show port: 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",
    "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 15.31. Show 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>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.

## 15.5.4. Update port

Method	URI	Description
PUT	/v2.0/ports/{port_id}	Updates the specified port.

**Normal response codes:** 200

### 15.5.4.1. 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 15.32. Update port: JSON request

```
{
  "port": {
    "network_id": "ee2d3158-3e80-4fb3-ba87-c99f515d85e7",
    "admin_state_up": true
  }
}
```

### 15.5.4.2. Response

#### Example 15.33. 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 15.34. 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.

## 15.6. Security groups and rules (security-groups)

Method	URI	Description
<b>GET</b>	/v2.0/security-groups	Lists all OpenStack Networking security groups to which the specified tenant has access.
<b>POST</b>	/v2.0/security-groups	Creates an OpenStack Networking security group.
<b>GET</b>	/v2.0/security-groups/{security_group_id}	Shows information for a specified security group.
<b>DELETE</b>	/v2.0/security-groups/{security_group_id}	Deletes an OpenStack Networking security group.
<b>GET</b>	/v2.0/security-group-rules	Lists a summary of all OpenStack Networking security group rules that the specified tenant can access.
<b>POST</b>	/v2.0/security-group-rules	Creates an OpenStack Networking security group rule.
<b>GET</b>	/v2.0/security-group-rules/{rules-security-groups-id}	Shows detailed information for a specified security group rule.
<b>DELETE</b>	/v2.0/security-group-rules/{rules-security-groups-id}	Deletes a specified rule from a OpenStack Networking security group.

## 15.6.1. List security groups

Method	URI	Description
<b>GET</b>	/v2.0/security-groups	Lists all OpenStack Networking security groups to which the specified tenant has access.

The list shows the unique ID for each security group and the rules that are associated with each security group.

**Normal response codes:** 200

**Error response codes:** unauthorized (401)

### 15.6.1.1. Request

#### Example 15.35. List security groups: JSON request

```
GET /v2.0/security-groups
Accept: application/json
```

This operation does not accept a request body.

### 15.6.1.2. Response

#### Example 15.36. 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"
                }
            ]
        }
    ]
}
```

```
        "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"
}
]
```

## 15.6.2. Create security group

Method	URI	Description
POST	/v2.0/security-groups	Creates an OpenStack Networking security group.

This operation creates a security group with default security group rules for the IPv4 and IPv6 ether types.

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401)

### 15.6.2.1. Request

#### Example 15.37. Create security group: JSON request

```
{
    "security_group": {
        "name": "new-webservers",
        "description": "security group for webservers"
    }
}
```

### 15.6.2.2. Response

#### Example 15.38. 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"
}
```

## 15.6.3. Show security group

Method	URI	Description
GET	/v2.0/security-groups/{security_group_id}	Shows information for a specified security group.

This operation returns a response body that contains the description, name, ID, and security group rules associated with the specified security group and tenant ID.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), itemNotFound (404)

### 15.6.3.1. Request

This table shows the URI parameters for the show security group request:

Name	Type	Description
{security_group_id}	Uuid	The unique identifier of the security group.

#### Example 15.39. Show security group: JSON request

```
GET /v2.0/security-groups/85cc3048-abc3-43cc-89b3-377341426ac5
Accept: application/json
```

This operation does not accept a request body.

### 15.6.3.2. Response

#### Example 15.40. Show security group: JSON response

```
{
  "security_group": {
    "description": "default",
    "id": "85cc3048-abc3-43cc-89b3-377341426ac5",
    "name": "default",
    "security_group_rules": [
      {
        "direction": "egress",
        "ethertype": "IPv6",
        "id": "3c0e45ff-adaf-4124-b083-bf390e5482ff",
        "port_range_max": null,
        "port_range_min": null,
        "protocol": null,
        "remote_group_id": null,
        "remote_ip_prefix": null,
        "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
      },
      {
        "direction": "egress",
        "ethertype": "IPv4",
        "id": "93aa42e5-80db-4581-9391-3a608bd0e448",
        "port_range_max": null,
        "port_range_min": null
      }
    ]
  }
}
```

```
        "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"
}
```

## 15.6.4. 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)

### 15.6.4.1. 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 15.41. Delete security group: JSON request

```
DELETE /v2.0/security-groups/e470bdfe-4869-459b-a561-cb3377efae59
Content-Type: application/json
Accept: application/json
```

This operation does not accept a request body.

### 15.6.4.2. Response

#### Example 15.42. Delete security group: JSON response

```
status: 204
```

This operation does not return a response body.

## 15.6.5. List security group rules

Method	URI	Description
<b>GET</b>	/v2.0/security-group-rules	Lists a summary of all OpenStack Networking security group rules that the specified tenant can access.

The list provides the unique ID for each security group rule.

**Normal response codes:** 200

**Error response codes:** unauthorized (401)

### 15.6.5.1. Request

#### Example 15.43. List security group rules: JSON request

```
GET /v2.0/security-group-rules/
Accept: application/json
```

This operation does not accept a request body.

### 15.6.5.2. Response

#### Example 15.44. List security group rules: JSON response

```
{
  "security_group_rules": [
    {
      "direction": "egress",
      "ethertype": "IPv6",
      "id": "3c0e45ff-adaf-4124-b083-bf390e5482ff",
      "port_range_max": null,
      "port_range_min": null,
      "protocol": null,
      "remote_group_id": null,
      "remote_ip_prefix": null,
      "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
      "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
    },
    {
      "direction": "egress",
      "ethertype": "IPv4",
      "id": "93aa42e5-80db-4581-9391-3a608bd0e448",
      "port_range_max": null,
      "port_range_min": null,
      "protocol": null,
      "remote_group_id": null,
      "remote_ip_prefix": null,
      "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
      "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
    },
    {
      "direction": "ingress",
      "ethertype": "IPv6",
      "id": "c0b09f00-1d49-4e64-a0a7-8a186d928138",
```

```
        "port_range_max": null,
        "port_range_min": null,
        "protocol": null,
        "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "remote_ip_prefix": null,
        "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
    },
    {
        "direction": "ingress",
        "ethertype": "IPv4",
        "id": "f7d45c89-008e-4bab-88ad-d6811724c51c",
        "port_range_max": null,
        "port_range_min": null,
        "protocol": null,
        "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "remote_ip_prefix": null,
        "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
    }
]
```

## 15.6.6. 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), buildInProgress (409)

### 15.6.6.1. Request

#### Example 15.45. Create security group rule: JSON request

```
{  
    "security_group_rule": {  
        "direction": "ingress",  
        "port_range_min": "80",  
        "ethertype": "IPv4",  
        "port_range_max": "80",  
        "protocol": "tcp",  
        "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",  
        "security_group_id": "a7734e61-b545-452d-a3cd-0189cbd9747a"  
    }  
}
```

### 15.6.6.2. Response

#### Example 15.46. Create security group rule: JSON response

```
{  
    "security_group_rule": {  
        "direction": "ingress",  
        "ethertype": "IPv4",  
        "id": "2bc0accf-312e-429a-956e-e4407625eb62",  
        "port_range_max": 80,  
        "port_range_min": 80,  
        "protocol": "tcp",  
        "remote_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",  
        "remote_ip_prefix": null,  
        "security_group_id": "a7734e61-b545-452d-a3cd-0189cbd9747a",  
        "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"  
    }  
}
```

## 15.6.7. Show security group rule

Method	URI	Description
GET	/v2.0/security-group-rules/{rules-security-groups-id}	Shows detailed information for a specified security group rule.

The response body contains the following information about the security group rule:

**Normal response codes:** 200

**Error response codes:** unauthorized (401), itemNotFound (404)

### 15.6.7.1. 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 15.47. Show security group rule: JSON request

```
GET /v2.0/security-group-rules/ 3c0e45ff-adaf-4124-b083-bf390e5482ff
Accept: application/json
```

This operation does not accept a request body.

### 15.6.7.2. Response

#### Example 15.48. Show security group rule: JSON response

```
{
  "security_group_rule": {
    "direction": "egress",
    "ethertype": "IPv6",
    "id": "3c0e45ff-adaf-4124-b083-bf390e5482ff",
    "port_range_max": null,
    "port_range_min": null,
    "protocol": null,
    "remote_group_id": null,
    "remote_ip_prefix": null,
    "security_group_id": "85cc3048-abc3-43cc-89b3-377341426ac5",
    "tenant_id": "e4f50856753b4dc6afee5fa6b9b6c550"
  }
}
```

## 15.6.8. Delete security group rule

Method	URI	Description
<b>DELETE</b>	/v2.0/security-group-rules/{rules-security-groups-id}	Deletes a specified rule from a OpenStack Networking security group.

**Normal response codes:** 204

**Error response codes:** unauthorized (401), itemNotFound (404)

### 15.6.8.1. Request

This table shows the URI parameters for the delete security group rule request:

Name	Type	Description
{rules-security-groups-id}	Uuid	The unique identifier of the security group rule.

#### Example 15.49. Delete security group rule: JSON request

```
DELETE /v2.0/security-group-rules/fc3c327a-b5b5-4cd3-9577-52893289ce08
Content-Type: application/json
Accept: application/json
```

This operation does not accept a request body.

### 15.6.8.2. Response

#### Example 15.50. Delete security group rule: JSON response

```
status: 204
```

This operation does not return a response body.

## 15.7. Layer-3 networking

Route packets between subnets, forward packets from internal networks to external ones, and access 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
<b>GET</b>	/v2.0/routers	Lists logical routers that are accessible to the tenant who submits the request.
<b>POST</b>	/v2.0/routers	Creates a logical router.

Method	URI	Description
<b>GET</b>	/v2.0/routers/{router_id}	Shows details for a specified router.
<b>PUT</b>	/v2.0/routers/{router_id}	Updates a logical router.
<b>DELETE</b>	/v2.0/routers/{router_id}	Deletes a logical router and, if present, its external gateway interface.
<b>PUT</b>	/v2.0/routers/{router_id}/add_router_interface	Adds an internal interface to a logical router.
<b>PUT</b>	/v2.0/routers/remove_router_interface	Removes an internal interface from a logical router.
<b>POST</b>	/v2.0/floatingips	Creates a floating IP, and, if you specify port information, associates the floating IP with an internal port.
<b>GET</b>	/v2.0/floatingips/{floatingip_id}	Shows details for a specified floating IP.
<b>PUT</b>	/v2.0/floatingips/{floatingip_id}	Updates a floating IP and its association with an internal port.
<b>DELETE</b>	/v2.0/floatingips/{floatingip_id}	Deletes a floating IP and, if present, its associated port.

## 15.7.1. 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](#) in the *OpenStack Networking API v2.0 Reference*.

**Normal response codes:** 200

**Error response codes:** unauthorized (401)

### 15.7.1.1. Request

This operation does not accept a request body.

### 15.7.1.2. Response

#### Example 15.51. List routers: JSON response

```
{
    "routers": [
        {
            "status": "ACTIVE",
            "external_gateway_info": null,
            "name": "second_routers",
            "admin_state_up": true,
            "tenant_id": "6b96ff0cb17a4b859e1e575d221683d3",
            "id": "7177abc4-5ae9-4bb7-b0d4-89e94a4abf3b"
        },
        {
            "status": "ACTIVE",
            "external_gateway_info": {
                "network_id": "3c5bcd3-6af9-4e6b-9c3e-c153e521cab8"
            },
            "name": "router1",
            "admin_state_up": true,
            "tenant_id": "33a40233088643acb66ff6eb0ebea679",
            "id": "a9254bdb-2613-4a13-ac4c-adc581fba50d"
        }
    ]
}
```

## 15.7.2. 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)

### 15.7.2.1. Request

#### Example 15.52. Create router: JSON request

```
{
  "router": {
    "name": "another_router",
    "external_gateway_info": {
      "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b"
    },
    "admin_state_up": true
  }
}
```

### 15.7.2.2. Response

#### Example 15.53. Create router: JSON response

```
{
  "router": {
    "status": "ACTIVE",
    "external_gateway_info": {
      "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b"
    },
    "name": "another_router",
    "admin_state_up": true,
    "tenant_id": "6b96ff0cb17a4b859e1e575d221683d3",
    "id": "8604a0de-7f6b-409a-a47c-a1cc7bc77b2e"
  }
}
```



## 15.7.3. Show router details

Method	URI	Description
GET	/v2.0/routers/{router_id}	Shows details for a specified 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](#) in the *OpenStack Networking API v2.0 Reference*.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.7.3.1. Request

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

Name	Type	Description
{router_id}	UUID	The UUID of the router.

This operation does not accept a request body.

### 15.7.3.2. Response

#### Example 15.54. Show router details: JSON response

```
{
    "router": {
        "status": "ACTIVE",
        "external_gateway_info": {
            "network_id": "85d76829-6415-48ff-9c63-5c5ca8c61ac6"
        },
        "name": "router1",
        "admin_state_up": true,
        "tenant_id": "d6554fe62e2f41efbb6e026fad5c1542",
        "id": "a07eea83-7710-4860-931b-5fe220fae533"
    }
}
```

## 15.7.4. Update router

Method	URI	Description
PUT	/v2.0/routers/{router_id}	Updates a logical router.

You can update the name, administrative state, and the external gateway. For more information about how to set the external gateway for a router, see the create router operation. This operation does not enable the update of router interfaces. To update a router, use the add router interface and remove router interface operations.

This example updates the external gateway information for a router:

```
PUT /v2.0/routers/{router_id}
Accept: application/json
```

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401), itemNotFound (404)

### 15.7.4.1. Request

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

Name	Type	Description
{router_id}	UUID	The UUID of the router.

#### Example 15.55. Update router: JSON request

```
{
    "router": {
        "external_gateway_info": {
            "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b"
        }
    }
}
```

### 15.7.4.2. Response

#### Example 15.56. Update router: JSON response

```
{
    "router": {
        "status": "ACTIVE",
        "external_gateway_info": {
            "network_id": "8ca37218-28ff-41cb-9b10-039601ea7e6b"
        },
        "name": "another_router",
        "admin_state_up": true,
        "tenant_id": "6b96ff0cb17a4b859e1e575d221683d3",
        "id": "8604a0de-7f6b-409a-a47c-a1cc7bc77b2e"
    }
}
```

## 15.7.5. Delete router

Method	URI	Description
<b>DELETE</b>	/v2.0/routers/{router_id}	Deletes a logical router and, if present, its external gateway interface.

This operation fails if the router has attached interfaces.

Use the remove router interface operation to remove all router interfaces before you delete the router.

This example deletes a router:

```
DELETE /v2.0/routers/{router_id}  
Accept: application/json
```

**Normal response codes:** 204

**Error response codes:** unauthorized (401), itemNotFound (404), conflict (409)

### 15.7.5.1. Request

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

Name	Type	Description
{router_id}	UUID	The UUID of the router.

This operation does not accept a request body.

## 15.7.6. Add interface to router

Method	URI	Description
PUT	/v2.0/routers/{router_id}/add_router_interface	Adds an internal interface to a logical router.

This operation attaches a subnet to an internal router interface. You must specify either a subnet or port ID in the request body. If you specify both IDs, the operation returns a 400 Bad Request error.

If you specify a subnet ID in the request body, the gateway IP address for the subnet is used to create the router interface.

If you specify a port ID in the request body, the IP address associated with the port is used to create the router interface.

The operation returns a 400 Bad Request error if several IP addresses are associated with the specified port, or if no IP address is associated with the port.

The operation returns a 409 Conflict error if the port is already used.

The port ID that is returned by this operation can either be the same ID passed in the request body or the ID of a new port created by this operation to attach the specified subnet to the router. After you run this operation, the device ID of this port is set to the router ID, and the device\_owner attribute is set to network:router\_interface, as shown in this example:

```
{
  "port": {
    "status": "ACTIVE",
    "name": "",
    "admin_state_up": true,
    "network_id": "5307648b-e836-4658-8f1a-ff7536870c64",
    "tenant_id": "6b96ff0cb17a4b859e1e575d221683d3",
    "device_owner": "network:router_interface",
    "mac_address": "fa:16:3e:f7:d1:9c",
    "fixed_ips": [
      {
        "subnet_id": "a2f1f29d-571b-4533-907f-5803ab96ead1",
        "ip_address": "10.1.1.1"
      }
    ],
    "id": "3a44f4e5-1694-493a-a1fb-393881c673a4",
    "device_id": "7177abc4-5ae9-4bb7-b0d4-89e94a4abf3b"
  }
}
```

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401), itemNotFound (404), conflict (409)

### 15.7.6.1. Request

This table shows the URI parameters for the add interface to router request:

Name	Type	Description
{router_id}	UUID	The UUID of the router.

**Example 15.57. Add interface to router: JSON request**

```
{  
    "subnet_id": "a2f1f29d-571b-4533-907f-5803ab96ead1"  
}
```

**15.7.6.2. Response****Example 15.58. Add interface to router: JSON response**

```
{  
    "subnet_id": "a2f1f29d-571b-4533-907f-5803ab96ead1",  
    "port_id": "3a44f4e5-1694-493a-a1fb-393881c673a4"  
}
```

## 15.7.7. Remove interface from router

Method	URI	Description
PUT	/v2.0/routers/ 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. You must specify either a subnet ID or port ID in the request body; this value is used to identify the 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, the operation returns a 409 Conflict error. The response contains information about the affected router and interface.

The operation returns a 404 Not Found if the router or the subnet and port do not exist or are not visible to you. As a consequence of this operation, the port connecting the router with the subnet is removed 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)

### 15.7.7.1. Request

#### Example 15.59. Remove interface from router: JSON request

```
{
    "subnet_id": "a2f1f29d-571b-4533-907f-5803ab96ead1"
}
```

### 15.7.7.2. Response

#### Example 15.60. 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"
}
```

## 15.7.8. 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.

If you do not specify port information in the request, you can issue an **PUT** request.

You can create floating IPs on external networks only. If you specify a network that is not external, such as `router:external=False`, the operation returns a 400 error.

If you do not specify a floating IP address in the request, the operation automatically allocates an address for the floating IP. If the requested floating IP address does not fall in the subnet range for the external network, the operation returns a 400 error. If the requested floating IP address is already in use, the operation returns a 409 error code.

You can associate the floating IP with an internal port by using the port ID attribute in the request body. If you specify a port ID that is not valid, the operation returns a 404 error code.

You must configure an IP address with the internal OpenStack Networking port associated with the floating IP or the operation returns a 400 error code. Because an OpenStack Networking port might be associated with multiple IP addresses, you can use the `fixed_ip_address` attribute in the request body to associate a particular IP address with the floating IP.

By default, this operation associates the floating IP with a single IP address that is configured on a port. Therefore, if a port has multiple IP addresses, you must specify the `fixed_ip_address` attribute.

If you specify an IP address that is not valid in the `fixed_ip_address` attribute, the operation returns a 400 error code. If the internal OpenStack Networking port and specified IP address are already associated with another floating IP, the operation returns a 409 error code.

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401), conflict (409)

### 15.7.8.1. Request

#### Example 15.61. Create floating IP: JSON request

```
{
    "floatingip": {
        "floating_network_id": "376da547-b977-4cfe-9cba-275c80deb57",
        "port_id": "ce705c24-clef-408a-bda3-7bbd946164ab"
    }
}
```

### 15.7.8.2. Response

#### Example 15.62. Create floating IP: JSON response

```
{
```

```
    "floatingip": {
        "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"
    }
}
```

## 15.7.9. Show floating IP details

Method	URI	Description
GET	/v2.0/floatingips/{floatingip_id}	Shows details for a specified floating IP.

Use the `fields` query parameter to control which fields are returned in the response body. For information, see [Filtering and Column Selection](#) in the *OpenStack Networking API v2.0 Reference*.

This example request shows details for a floating IP in JSON format. This example also filters the result by the `fixed_ip_address` and `floating_ip_address` fields.

```
GET /v2.0/floatingips/{floatingip_id}?fields=fixed_ip_address&fields=\
floating_ip_address
Accept: application/json
```

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.7.9.1. 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.

### 15.7.9.2. Response

#### Example 15.63. Show floating IP details: JSON response

```
{
    "floatingip": {
        "fixed_ip_address": "10.0.0.3",
        "floating_ip_address": "172.24.4.228"
    }
}
```

## 15.7.10. Update floating IP

Method	URI	Description
PUT	/v2.0/floatingips/{floatingip_id}	Updates a floating IP and its association with an internal port.

The association process is the same as the process for the create floating IP operation.

To disassociate a floating IP from a port, set the `port_id` attribute to null or omit it from the request body.

This example updates a floating IP:

```
PUT /v2.0/floatingips/{floatingip_id}
Accept: application/json
```

Depending on the request body that you submit, this request associates a port with or disassociates a port from a floating IP.

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401), itemNotFound (404), conflict (409)

### 15.7.10.1. 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 15.64. Update floating IP (associate port): JSON

```
{
  "floatingip": {
    "port_id": "fc861431-0e6c-4842-a0ed-e2363f9bc3a8"
  }
}
```

#### Example 15.65. Update floating IP (disassociate port): JSON

```
{
  "floatingip": {
    "port_id": null
  }
}
```

### 15.7.10.2. Response

#### Example 15.66. Update floating IP (associate port): JSON

```
{
  "floatingip": {
    "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
```

```
        "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
        "floating_network_id": "376da547-b977-4cfe-9cba-275c80deb57",
        "fixed_ip_address": "10.0.0.4",
        "floating_ip_address": "172.24.4.228",
        "port_id": "fc861431-0e6c-4842-a0ed-e2363f9bc3a8",
        "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7"
    }
}
```

### Example 15.67. Update floating IP (disassociate port): JSON

```
{
    "floatingip": {
        "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
        "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
        "floating_network_id": "376da547-b977-4cfe-9cba-275c80deb57",
        "fixed_ip_address": null,
        "floating_ip_address": "172.24.4.228",
        "port_id": null,
        "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7"
    }
}
```

## 15.7.11. Delete floating IP

Method	URI	Description
<b>DELETE</b>	/v2.0/floatingips/{floatingip_id}	Deletes a floating IP and, if present, its associated port.

This example deletes a floating IP:

```
DELETE /v2.0/floatingips/{floatingip_id}
Accept: application/json
```

**Normal response codes:** 204

**Error response codes:** unauthorized (401), itemNotFound (404)

### 15.7.11.1. 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.

## 15.8. Metering labels and rules

Create, modify, and delete OpenStack Layer3 Metering labels and rules.

Method	URI	Description
<b>GET</b>	/v2.0/metering-labels	Lists all I3 metering labels that belong to the specified tenant.
<b>POST</b>	/v2.0/metering-labels	Creates a I3 metering label.
<b>GET</b>	/v2.0/metering-labels/{metering_label_id}	Shows informations for a specified metering label.
<b>DELETE</b>	/v2.0/metering-labels/{metering_label_id}	Deletes a I3 metering label.
<b>GET</b>	/v2.0/metering-label-rules	Lists a summary of all I3 metering label rules belonging to the specified tenant.
<b>POST</b>	/v2.0/metering-label-rules	Creates a I3 metering label rule.
<b>GET</b>	/v2.0/metering-label-rules/{metering-label-rule-id}	Shows detailed informations for a specified metering label rule.
<b>DELETE</b>	/v2.0/metering-label-rules/{metering-label-rule-id}	Deletes a specified I3 metering label rule.

## 15.8.1. List metering labels

Method	URI	Description
GET	/v2.0/metering-labels	Lists all I3 metering labels that belong to the specified tenant.

The list includes the unique ID for each metering labels.

This operation does not require a request body.

This operation returns a response body.

**Normal response codes:** 200

**Error response codes:** unauthorized (401)

### 15.8.1.1. Request

#### Example 15.68. List metering labels: JSON request

```
GET /v2.0/metering/metering-labels HTTP/1.1
Host: controlnode:9696
User-Agent: python-neutronclient
Content-Type: application/json
Accept: application/json
X-Auth-Token: c52a1b304fec4ca0ac85dc1741eec6e2
```

This operation does not accept a request body.

### 15.8.1.2. Response

#### Example 15.69. List metering labels: JSON response

```
{
    "metering_labels": [
        {
            "tenant_id": "45345b0ee1ea477fac0f541b2cb79cd4",
            "description": "label1 description",
            "name": "label1",
            "id": "a6700594-5b7a-4105-8bfe-723b346ce866"
        },
        {
            "tenant_id": "45345b0ee1ea477fac0f541b2cb79cd4",
            "description": "label2 description",
            "name": "label2",
            "id": "e131d186-b02d-4c0b-83d5-0c0725c4f812"
        }
    ]
}
```

## 15.8.2. Create metering label

Method	URI	Description
POST	/v2.0/metering-labels	Creates a I3 metering label.

This operation requires a request body.

The following table describes the required and optional attributes in the request body:

**Table 15.1. Create Metering label rule Request Attributes**

Attribute	Required	Description
name	Required	The name of the metering label.
description	Optional	Description for the metering label.

This operation returns a response body, which contains the following informations about the metering label:

- name. Name of the metering label.
- description. Description of the metering label.
- tenant\_id. The tenant ID for the specified metering label.
- id. The metering label ID

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401)

### 15.8.2.1. Request

#### Example 15.70. Create metering label: JSON request

```
{
  "metering_label": {
    "name": "label1",
    "description": "description of label1"
  }
}
```

### 15.8.2.2. Response

#### Example 15.71. Create metering label: JSON response

```
{
  "metering_label": {
    "tenant_id": "45345b0ee1ea477fac0f541b2cb79cd4",
    "description": "description of label1",
    "name": "label1",
    "id": "bc91b832-8465-40a7-a5d8-ba87de442266"
  }
}
```

```
    }  
}
```

## 15.8.3. Show metering label

Method	URI	Description
GET	/v2.0/metering-labels/{metering_label_id}	Shows informations for a specified metering label.

This operation does not require a request body.

This operation returns a response body that contains the description, name, ID.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), itemNotFound (404)

### 15.8.3.1. Request

This table shows the URI parameters for the show metering label request:

Name	Type	Description
{metering_label_id}	Uuid	The unique identifier of the metering label.

#### Example 15.72. 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.

### 15.8.3.2. Response

#### Example 15.73. Show metering label: JSON response

```
{
    "metering_label": {
        "tenant_id": "45345b0ee1ea477fac0f541b2cb79cd4",
        "description": "label1 description",
        "name": "label1",
        "id": "a6700594-5b7a-4105-8bfe-723b346ce866"
    }
}
```

## 15.8.4. Delete metering label

Method	URI	Description
<b>DELETE</b>	/v2.0/metering-labels/{metering_label_id}	Deletes a I3 metering label.

This operation deletes a I3 metering label.

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)

### 15.8.4.1. 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 15.74. Delete metering label: JSON request

```
DELETE /v2.0/metering/metering-labels/a6700594-5b7a-4105-8bfe-723b346ce866
HTTP/1.1
Host: controlnode:9696
User-Agent: python-neutronclient
Content-Type: application/json
Accept: application/json
X-Auth-Token: c52a1b304fec4ca0ac85dc1741eec6e2
```

This operation does not accept a request body.

### 15.8.4.2. Response

#### Example 15.75. Delete metering label: JSON response

```
status: 204
```

This operation does not return a response body.

## 15.8.5. List metering label rules

Method	URI	Description
GET	/v2.0/metering-label-rules	Lists a summary of all I3 metering label rules belonging to the specified tenant.

The list provides the unique ID for each metering label rule.

This operation does not require a request body. This operation returns a response body.

**Normal response codes:** 200

**Error response codes:** unauthorized (401)

### 15.8.5.1. Request

#### Example 15.76. List metering label rules: JSON request

```
GET /v2.0/metering/metering-label-rules HTTP/1.1
Host: controlnode:9696
User-Agent: python-neutronclient
Content-Type: application/json
Accept: application/json
X-Auth-Token: c52a1b304fec4ca0ac85dc1741eec6e2
```

This operation does not accept a request body.

### 15.8.5.2. Response

#### Example 15.77. List metering label rules: JSON response

```
{
    "metering_label_rules": [
        {
            "remote_ip_prefix": "20.0.0.0/24",
            "direction": "ingress",
            "metering_label_id": "e131d186-b02d-4c0b-83d5-0c0725c4f812",
            "id": "9536641a-7d14-4dc5-afaf-93a973ce0eb8",
            "excluded": false
        },
        {
            "remote_ip_prefix": "10.0.0.0/24",
            "direction": "ingress",
            "metering_label_id": "e131d186-b02d-4c0b-83d5-0c0725c4f812",
            "id": "ffc6fd15-40de-4e7d-b617-34d3f7a93aec",
            "excluded": false
        }
    ]
}
```

## 15.8.6. Create metering label rule

Method	URI	Description
POST	/v2.0/metering-label-rules	Creates a l3 metering label rule.

This operation requires a request body.

The following table describes the required and optional attributes in the request body:

**Table 15.2. Create Metering label rule Request Attributes**

Attribute	Required	Description
direction	Optional	Ingress or egress: The direction in which metering rule is applied. Default: ingress
metering_label_id	Required	The meteting label ID to associate with this metering rule.
excluded	Optional	Specify whether the remote_ip_prefix will be excluded or not from traffic counters of the metering label, ie: to not count the traffic of a specific IP address or a range. Default: False
remote_ip_prefix	Required	The remote IP prefix to be associated with this metering rule. packet.

This operation returns a response body.

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401), itemNotFound (404), buildIn-Progress (409)

### 15.8.6.1. Request

#### Example 15.78. Create metering label rule: JSON request

```
{
    "metering_label_rule": {
        "remote_ip_prefix": "10.0.1.0/24",
        "direction": "ingress",
        "metering_label_id": "e131d186-b02d-4c0b-83d5-0c0725c4f812"
    }
}
```

### 15.8.6.2. Response

#### Example 15.79. Create metering label rule: JSON response

```
{
    "metering_label_rule": {
        "remote_ip_prefix": "10.0.1.0/24",
```

```
        "direction": "ingress",
        "metering_label_id": "e131d186-b02d-4c0b-83d5-0c0725c4f812",
        "id": "00e13b58-b4f2-4579-9c9c-7ac94615f9ae",
        "excluded": false
    }
```

## 15.8.7. Show metering label rule

Method	URI	Description
GET	/v2.0/metering-label-rules/{metering-label-rule-id}	Shows detailed informations for a specified metering label rule.

This operation does not require a request body.

This operation returns a response body, which contains the following informations about the metering label rule:

- **direction**. Either ingress or egress.
- **excluded**. Either True or False.
- The ID for the specified metering label rule
- The remote IP prefix
- The metering label ID for the metering label with which the rule is associated

**Normal response codes:** 200

**Error response codes:** unauthorized (401), itemNotFound (404)

### 15.8.7.1. 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 15.80. 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.

### 15.8.7.2. Response

#### Example 15.81. 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
    }
}
```

## 15.8.8. Delete metering label rule

Method	URI	Description
<b>DELETE</b>	/v2.0/metering-label-rules/{metering-label-rule-id}	Deletes a specified I3 metering label rule.

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)

### 15.8.8.1. 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 15.82. Delete metering label rule: JSON request

```
DELETE /v2.0/metering/metering-labels/37b31179-71ee-4f0a-b130-0eeb28e7ede7
HTTP/1.1
Host: controlnode:9696
User-Agent: python-neutronclient
Content-Type: application/json
Accept: application/json
X-Auth-Token: c52a1b304fec4ca0ac85dc1741eec6e2
```

This operation does not accept a request body.

### 15.8.8.2. Response

#### Example 15.83. Delete metering label rule: JSON response

```
status: 204
```

This operation does not return a response body.

## 15.9. Load-Balancer-as-a-Service (LBaaS)

The LBaaS extension enables OpenStack tenants to load-balance their VM traffic. The extension enables you to load-balance client traffic from one network to application services, such as VMs, on the same or a different network. You can load-balance several protocols, such as TCP and HTTP and monitor the health of application services. The LBaaS extension supports session persistence.

Method	URI	Description
<b>POST</b>	/v2.0/loadbalancers	Creates a load balancer.

Method	URI	Description
<b>GET</b>	/v2.0/loadbalancers	Lists load balancers.
<b>GET</b>	/v2.0/loadbalancers/{load_balancer_id}	Shows the configuration details for a specified load balancer.
<b>PUT</b>	/v2.0/loadbalancers/{load_balancer_id}	Updates the editable attributes of a specified load balancer
<b>DELETE</b>	/v2.0/loadbalancers/{load_balancer_id}	Deletes a load balancer.
<b>POST</b>	/v2.0/listeners	Creates a listener.
<b>GET</b>	/v2.0/listeners	Lists listeners.
<b>GET</b>	/v2.0/listeners/{listener_id}	Shows the configuration details for a specified listener.
<b>PUT</b>	/v2.0/listeners/{listener_id}	Updates editable attributes of a specified listener.
<b>DELETE</b>	/v2.0/listeners/{listener_id}	Deletes an orphaned listener.
<b>POST</b>	/v2.0/pools	Creates a load balancer pool.
<b>GET</b>	/v2.0/pools	Lists pools.
<b>GET</b>	/v2.0/pools/{pool_id}	Shows the configuration details for a specified pool.
<b>PUT</b>	/v2.0/pools/{pool_id}	Updates editable attributes of a specified pool.
<b>DELETE</b>	/v2.0/pools/{pool_id}	Deletes an orphaned pool.
<b>POST</b>	/v2.0/pools/{pool_id}/members	Creates a load balancer member.
<b>GET</b>	/v2.0/pools/{pool_id}/members	Lists members in a specified pool.
<b>GET</b>	/v2.0/pools/{pool_id}/members/{member_id}	Shows the configuration details for a specified member.
<b>PUT</b>	/v2.0/pools/{pool_id}/members/{member_id}	Updates a specified load balancer member.
<b>DELETE</b>	/v2.0/pools/{pool_id}/members/{member_id}	Deletes a specified load balancer member.
<b>POST</b>	/v2.0/healthmonitors	Creates a health monitor.
<b>GET</b>	/v2.0/healthmonitors	Lists health monitors.
<b>GET</b>	/v2.0/healthmonitors/{health_monitor_id}	Shows the configuration details for a specified health monitor.
<b>PUT</b>	/v2.0/healthmonitors/{health_monitor_id}	Updates the editable attributes of a specified health monitor.
<b>DELETE</b>	/v2.0/healthmonitors/{health_monitor_id}	Deletes an orphaned health monitor.

## 15.9.1. Create load balancer

Method	URI	Description
POST	/v2.0/loadbalancers	Creates a load balancer.

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401)

### 15.9.1.1. Request

#### Example 15.84. Create load balancer: JSON request

```
{  
    "loadbalancer": {  
        "name": "Example LB",  
        "description": "A very simple example load balancer.",  
        "vip_subnet": "SUBNET_ID",  
        "tenant_id": "7725fe12-1c14-4f45-ba8e-44bf01763578"  
    }  
}
```

### 15.9.1.2. Response

#### Example 15.85. Create load balancer: JSON response

```
{  
    "loadbalancer": {  
        "id": "8992a43f-83af-4b49-9af9-c2bfb82d7d7",  
        "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",  
        "status": "PENDING_CREATE"  
    }  
}
```

## 15.9.2. List load balancers

Method	URI	Description
GET	/v2.0/loadbalancers	Lists load balancers.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.2.1. Request

This operation does not accept a request body.

### 15.9.2.2. Response

#### Example 15.86. List load balancers: JSON response

```
{  
    "loadbalancers": {  
        "id": "3b98602c-3cfe-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"  
    }  
}
```

## 15.9.3. Show load balancer configuration details

Method	URI	Description
GET	/v2.0/loadbalancers/{load_balancer_id}	Shows the configuration details for a specified load balancer.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.3.1. Request

This operation does not accept a request body.

### 15.9.3.2. Response

#### Example 15.87. Show load balancer configuration details: JSON response

```
{  
    "loadbalancer": {  
        "id": "8992a43f-83af-4b49-9afd-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"  
    }  
}
```

## 15.9.4. Update load balancer

Method	URI	Description
PUT	/v2.0/loadbalancers/{load_balancer_id}	Updates the editable attributes of a specified load balancer

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.4.1. Request

#### Example 15.88. Update load balancer: JSON request

```
{  
    "loadbalancer":{  
        "name":"A New Example LB",  
        "description":"A new very simple example load balancer."  
    }  
}
```

### 15.9.4.2. Response

#### Example 15.89. Update load balancer: JSON response

```
{  
    "loadbalancer":{  
        "id":"8992a43f-83af-4b49-9af8-c2bfbd82d7d7",  
        "name":"A New Example LB",  
        "description":"A new 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"  
    }  
}
```

## 15.9.5. Delete load balancer

Method	URI	Description
<b>DELETE</b>	/v2.0/loadbalancers/{load_balancer_id}	Deletes a load balancer.

Upon successful deletion, if the deleted load balancer used any listeners, they are detached but not deleted.

**Normal response codes:** 204

**Error response codes:** unauthorized (401), itemNotFound (404), conflict (409)

### 15.9.5.1. Request

This operation does not accept a request body.

### 15.9.5.2. Response

#### Example 15.90. Delete load balancer: JSON response

```
{  
    "loadbalancer":{  
        "active_connections":0,  
        "total_connections":0,  
        "bytes_in":0,  
        "bytes_out":0  
    }  
}
```

## 15.9.6. Create listener

Method	URI	Description
POST	/v2.0/listeners	Creates a listener.

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401)

### 15.9.6.1. Request

#### Example 15.91. Create listener: JSON request

```
{  
    "listener": {  
        "default_pool_id": "8311446e-8a13-4c00-95b3-03a92f9759c7",  
        "name": "Example HTTPS Listener",  
        "description": "A very simple example of an HTTPS listener.",  
        "tenant_id": "352686b7-c4b2-44ec-a458-84239713f685",  
        "connection_limit": 200,  
        "protocol": "https",  
        "protocol_port": 443,  
        "load_balancer_id": "b8a35470-f65d-11e3-a3ac-0800200c9a66"  
    }  
}
```

### 15.9.6.2. Response

#### Example 15.92. Create listener: JSON response

```
{  
    "listener": {  
        "id": "9bb09b2b-a0e3-4fa7-8020-76efd4b30d30",  
        "default_pool_id": "8311446e-8a13-4c00-95b3-03a92f9759c7",  
        "tenant_id": "352686b7-c4b2-44ec-a458-84239713f685",  
        "name": "Example HTTPS Listener",  
        "description": "A very simple example of an HTTPS listener",  
        "protocol": "https",  
        "protocol_port": 443,  
        "load_balancer_id": "b8a35470-f65d-11e3-a3ac-0800200c9a66",  
        "admin_state_up": true,  
        "status": "ACTIVE"  
    }  
}
```

## 15.9.7. List listeners

Method	URI	Description
GET	/v2.0/listeners	Lists listeners.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.7.1. Request

This operation does not accept a request body.

### 15.9.7.2. Response

#### Example 15.93. List listeners: JSON response

```
{
  "listeners": [
    {
      "id": "fddd4b15-faca-4a60-af4c-43317b9ad363",
      "tenant_id": "783b31af-6635-48b2-a807-091d9973e3a9",
      "name": "Example HTTPS Listener",
      "description": "A very simple example of an HTTPS listener.",
      "protocol": "https",
      "protocol_port": "443",
      "load_balancer_id": "b8a35470-f65d-11e3-a3ac-0800200c9a66",
      "admin_state_up": true,
      "status": "ACTIVE"
    },
    {
      "id": "9341aafc-7e7d-4807-8199-9e81dadd0577",
      "tenant_id": "783b31af-6635-48b2-a807-091d9973e3a9",
      "name": "Example HTTPS Listener",
      "description": "A very simple example of an HTTPS listener.",
      "protocol": "https",
      "protocol_port": "443",
      "load_balancer_id": "b8a35470-f65d-11e3-a3ac-0800200c9a66",
      "admin_state_up": true,
      "status": "ACTIVE"
    }
  ]
}
```

## 15.9.8. Show listener configuration details

Method	URI	Description
GET	/v2.0/listeners/{listener_id}	Shows the configuration details for a specified listener.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.8.1. Request

This operation does not accept a request body.

### 15.9.8.2. Response

#### Example 15.94. Show listener configuration details: JSON response

```
{
  "listener": {
    "id": "9bb09b2b-a0e3-4fa7-8020-76efd4b30d30",
    "default_pool_id": "8311446e-8a13-4c00-95b3-03a92f9759c7",
    "tenant_id": "352686b7-c4b2-44ec-a458-84239713f685",
    "name": "Example HTTPS Listener",
    "description": "A very simple example of an HTTPS listener",
    "protocol": "https",
    "protocol_port": "443",
    "load_balancer_id": "b8a35470-f65d-11e3-a3ac-0800200c9a66",
    "admin_state_up": true,
    "status": "ACTIVE"
  }
}
```

## 15.9.9. Update listener

Method	URI	Description
PUT	/v2.0/listeners/{listener_id}	Updates editable attributes of a specified listener.

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.9.1. Request

#### Example 15.95. Update listener: JSON request

```
{  
    "listener":{  
        "name":"A New Example HTTPS Listener"  
    }  
}
```

### 15.9.9.2. Response

#### Example 15.96. Update listener: JSON response

```
{  
    "listener":{  
        "id":"9bb09b2b-a0e3-4fa7-8020-76efd4b30d30",  
        "default_pool_id":"8311446e-8a13-4c00-95b3-03a92f9759c7",  
        "tenant_id":"352686b7-c4b2-44ec-a458-84239713f685",  
        "name":"A New Example HTTPS Listener",  
        "description":"A very simple example of an HTTPS listener.",  
        "protocol":"https",  
        "protocol_port":443,  
        "load_balancer_id":"b8a35470-f65d-11e3-a3ac-0800200c9a66",  
        "admin_state_up":true,  
        "status":ACTIVE  
    }  
}
```

## 15.9.10. Delete listener

Method	URI	Description
<b>DELETE</b>	/v2.0/listeners/{listener_id}	Deletes an orphaned listener.

Returns an error if any load balancers still use the listener.

Upon successful deletion, if the deleted listener used any pools, they are detached but not deleted.

**Normal response codes:** 204

**Error response codes:** unauthorized (401), itemNotFound (404), conflict (409)

### 15.9.10.1. Request

This operation does not accept a request body.

## 15.9.11. Create load balancer pool

Method	URI	Description
POST	/v2.0/pools	Creates a load balancer pool.

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401)

### 15.9.11.1. Request

#### Example 15.97. Create load balancer pool: JSON request

```
{  
    "pool": {  
        "name": "Example pool",  
        "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",  
        "protocol": "HTTP",  
        "lb_algorithm": "ROUND_ROBIN"  
    }  
}
```

### 15.9.11.2. Response

#### Example 15.98. Create load balancer pool: JSON response

```
{  
    "pool": {  
        "id": "332abe93-f488-41ba-870b-2ac66be7f853",  
        "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",  
        "name": "Example pool",  
        "description": "",  
        "protocol": "tcp",  
        "lb_algorithm": "ROUND_ROBIN",  
        "session_persistence": {  
        },  
        "healthmonitor_id": null,  
        "members": [  
        ],  
        "admin_state_up": true,  
        "status": "ACTIVE"  
    }  
}
```

## 15.9.12. List pools

Method	URI	Description
GET	/v2.0/pools	Lists pools.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.12.1. Request

This operation does not accept a request body.

### 15.9.12.2. Response

#### Example 15.99. List pools: JSON response

```
{
  "pools": [
    {
      "id": "332abe93-f488-41ba-870b-2ac66be7f853",
      "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",
      "name": "Example pool",
      "description": "",
      "protocol": "tcp",
      "lb_algorithm": "ROUND_ROBIN",
      "session_persistence": {},
      "healthmonitor_id": null,
      "members": [],
      "admin_state_up": true,
      "status": "ACTIVE"
    },
    {
      "id": "332abe93-f488-41ba-870b-2ac66be7f854",
      "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",
      "name": "Example pool",
      "description": "",
      "protocol": "tcp",
      "lb_algorithm": "ROUND_ROBIN",
      "session_persistence": {},
      "healthmonitor_id": null,
      "members": [],
      "admin_state_up": true,
      "status": "ACTIVE"
    }
  ]
}
```

## 15.9.13. Show pool configuration details

Method	URI	Description
GET	/v2.0/pools/{pool_id}	Shows the configuration details for a specified pool.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.13.1. Request

This operation does not accept a request body.

### 15.9.13.2. Response

#### Example 15.100. Show pool configuration details: JSON response

```
{
  "pool": {
    "id": "332abe93-f488-41ba-870b-2ac66be7f853",
    "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",
    "name": "Example pool",
    "description": "",
    "protocol": "tcp",
    "lb_algorithm": "ROUND_ROBIN",
    "session_persistence": {},
    "healthmonitor_id": null,
    "members": [],
    "admin_state_up": true,
    "status": "ACTIVE"
  }
}
```

## 15.9.14. Update pool

Method	URI	Description
PUT	/v2.0/pools/{pool_id}	Updates editable attributes of a specified pool.

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.14.1. Request

#### Example 15.101. Update pool: JSON request

```
{  
    "pool":{  
        "lb_algorithm": "LEAST_CONNECTIONS",  
        "name": "New pool name"  
    }  
}
```

### 15.9.14.2. Response

#### Example 15.102. Update pool: JSON response

```
{  
    "pool":{  
        "id": "332abe93-f488-41ba-870b-2ac66be7f853",  
        "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",  
        "name": "New pool name",  
        "description": "",  
        "protocol": "tcp",  
        "lb_algorithm": "LEAST_CONNECTIONS",  
        "session_persistence": {  
        },  
        "healthmonitor_id": null,  
        "members": [  
        ],  
        "admin_state_up": true,  
        "status": "ACTIVE"  
    }  
}
```

## 15.9.15. Delete pool

Method	URI	Description
<b>DELETE</b>	/v2.0/pools/{pool_id}	Deletes an orphaned pool.

Returns an error if any listeners still use the pool.

Upon successful deletion, if the deleted pool used any child primitives, they are detached but not deleted.

**Normal response codes:** 204

**Error response codes:** unauthorized (401), itemNotFound (404), conflict (409)

### 15.9.15.1. Request

This operation does not accept a request body.

## 15.9.16. Create load balancer member

Method	URI	Description
POST	/v2.0/pools/{pool_id}/members	Creates a load balancer member.

Members are individual back-end services that are being load balanced. Typically, these are web application servers. They are represented as a pool, IP address, and Layer 4 port tuple.

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401)

### 15.9.16.1. Request

#### Example 15.103. Create load balancer member: JSON request

```
{  
  "member": {  
    "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",  
    "address": "192.0.2.14",  
    "protocol_port": 8080,  
    "subnet_id": "SUBNET_ID"  
  }  
}
```

### 15.9.16.2. Response

#### Example 15.104. Create load balancer member: JSON response

```
{  
  "member": {  
    "id": "975592ca-e308-48ad-8298-731935ee9f45",  
    "address": "192.0.2.14",  
    "protocol_port": 8080,  
    "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",  
    "admin_state_up": true,  
    "weight": 1,  
    "subnet_id": "SUBNET_ID",  
    "status": "DOWN"  
  }  
}
```

## 15.9.17. List pool members

Method	URI	Description
GET	/v2.0/pools/{pool_id}/members	Lists members in a specified pool.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.17.1. Request

This operation does not accept a request body.

### 15.9.17.2. Response

#### Example 15.105. List pool members: JSON response

```
{  
    "members": {  
        "id": "975592ca-e308-48ad-8298-731935ee9f45",  
        "address": "192.0.2.14",  
        "protocol_port": 8080,  
        "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",  
        "admin_state_up": true,  
        "weight": 1,  
        "subnet_id": "SUBNET_ID",  
        "status": "DOWN"  
    }  
}
```

## 15.9.18. Show member configuration details

Method	URI	Description
GET	/v2.0/pools/{pool_id}/members/{member_id}	Shows the configuration details for a specified member.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.18.1. Request

This operation does not accept a request body.

### 15.9.18.2. Response

#### Example 15.106. Show member configuration details: JSON response

```
{  
    "member": {  
        "id": "975592ca-e308-48ad-8298-731935ee9f45",  
        "address": "192.0.2.14",  
        "protocol_port": 8080,  
        "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",  
        "admin_state_up": true,  
        "weight": 1,  
        "subnet_id": "SUBNET_ID",  
        "status": "DOWN"  
    }  
}
```

## 15.9.19. Update member

Method	URI	Description
PUT	/v2.0/pools/{pool_id}/members/{member_id}	Updates a specified load balancer member.

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.19.1. Request

#### Example 15.107. Update member: JSON request

```
{  
    "member": {  
        "admin_state_up": false  
    }  
}
```

### 15.9.19.2. Response

#### Example 15.108. Update member: JSON response

```
{  
    "member": {  
        "id": "975592ca-e308-48ad-8298-731935ee9f45",  
        "address": "192.0.2.14",  
        "protocol_port": 8080,  
        "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",  
        "admin_state_up": false,  
        "weight": 1,  
        "subnet_id": "SUBNET_ID",  
        "status": "DOWN"  
    }  
}
```

## 15.9.20. Delete member

Method	URI	Description
<b>DELETE</b>	/v2.0/pools/{pool_id}/members/{member_id}	Deletes a specified load balancer member.

**Normal response codes:** 204

**Error response codes:** unauthorized (401), itemNotFound (404), conflict (409)

### 15.9.20.1. Request

This operation does not accept a request body.

## 15.9.21. Create load balancer health monitor

Method	URI	Description
POST	/v2.0/healthmonitors	Creates a health monitor.

**Normal response codes:** 201

**Error response codes:** badRequest (400), unauthorized (401)

### 15.9.21.1. Request

#### Example 15.109. Create load balancer health monitor: JSON request

```
{  
    "healthmonitor": {  
        "type": "HTTP",  
        "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",  
        "delay": 20,  
        "timeout": 10,  
        "max_retries": 5,  
        "url_path": "/check",  
        "expected_codes": "200-299"  
    }  
}
```

### 15.9.21.2. Response

#### Example 15.110. Create load balancer health monitor: JSON response

```
{  
    "healthmonitor": {  
        "id": "f3eeab00-8367-4524-b662-55e64d4cacb5",  
        "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",  
        "type": "HTTP",  
        "delay": 20,  
        "timeout": 10,  
        "max_retries": 5,  
        "http_method": "GET",  
        "url_path": "/check",  
        "expected_codes": "200-299",  
        "admin_state_up": true,  
        "status": "ACTIVE"  
    }  
}
```

## 15.9.22. List health monitors

Method	URI	Description
GET	/v2.0/healthmonitors	Lists health monitors.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.22.1. Request

This operation does not accept a request body.

### 15.9.22.2. Response

#### Example 15.111. List health monitors: JSON response

```
{  
    "healthmonitors": {  
        "id": "f3eeab00-8367-4524-b662-55e64d4cacb5",  
        "tenant_id": "453105b9-1754-413f-aab1-55flaf620750",  
        "type": "HTTP",  
        "delay": 20,  
        "timeout": 10,  
        "max_retries": 5,  
        "http_method": "GET",  
        "url_path": "/check",  
        "expected_codes": "200-299",  
        "admin_state_up": true,  
        "status": "ACTIVE"  
    }  
}
```

## 15.9.23. Show health monitor configuration details

Method	URI	Description
GET	/v2.0/healthmonitors/{health_monitor_id}	Shows the configuration details for a specified health monitor.

**Normal response codes:** 200

**Error response codes:** unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.23.1. Request

This operation does not accept a request body.

### 15.9.23.2. Response

#### Example 15.112. Show health monitor configuration details: JSON response

```
{
  "healthmonitor": {
    "id": "f3eeab00-8367-4524-b662-55e64d4cacb5",
    "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
    "type": "HTTP",
    "delay": 20,
    "timeout": 10,
    "max_retries": 5,
    "http_method": "GET",
    "url_path": "/check",
    "expected_codes": "200-299",
    "admin_state_up": true,
    "status": "ACTIVE"
  }
}
```

## 15.9.24. Update health monitor

Method	URI	Description
PUT	/v2.0/healthmonitors/{health_monitor_id}	Updates the editable attributes of a specified health monitor.

**Normal response codes:** 200

**Error response codes:** badRequest (400), unauthorized (401), forbidden (403), itemNotFound (404)

### 15.9.24.1. Request

#### Example 15.113. Update health monitor: JSON request

```
{  
    "healthmonitor": {  
        "path": "/healthcheck",  
        "delay": 3  
    }  
}
```

### 15.9.24.2. Response

#### Example 15.114. Update health monitor: JSON response

```
{  
    "healthmonitor": {  
        "id": "f3eeab00-8367-4524-b662-55e64d4cacb5",  
        "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",  
        "type": "HTTP",  
        "delay": 3,  
        "timeout": 10,  
        "max_retries": 5,  
        "http_method": "GET",  
        "url_path": "/healthcheck",  
        "expected_codes": "200-299",  
        "admin_state_up": true,  
        "status": "ACTIVE"  
    }  
}
```

## 15.9.25. Delete health monitor

Method	URI	Description
<b>DELETE</b>	/v2.0/healthmonitors/{health_monitor_id}	Deletes an orphaned health monitor.

Returns an error if any pools still use the health monitor.

**Normal response codes:** 204

**Error response codes:** unauthorized (401), itemNotFound (404), conflict (409)

### 15.9.25.1. Request

This operation does not accept a request body.

# 16. Object Storage API v1 (SUPPORTED)

Manage 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](#).

## 16.1. Accounts

List containers for a specified account. Create, update, show, and delete account metadata.

Method	URI	Description
<b>GET</b>	/v1/{account}{?limit,marker,end_marker,format,prefix,delim-iter}	Shows details for a specified account and lists containers, sorted by name, in the account.
<b>POST</b>	/v1/{account}	Creates, updates, or deletes account metadata.
<b>HEAD</b>	/v1/{account}	Shows metadata for a specified account.

## 16.1.1. Show account details and list containers

Method	URI	Description
GET	/v1/{account}{?limit,marker,end_marker,format,prefix,delim-iter}	Shows details for a specified 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.

This operation does not accept a request body.

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"
```

See the example response below.

- List containers and ask for an XML response: `curl -i $publicURL?format=xml -X GET -H "X-Auth-Token: $token"`

See the example response below.

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.

If the request succeeds, the operation returns one of these status codes:

- 200. Success. The response body lists the containers.
- 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 parameters, and you have reached the end of the list.

**Normal response codes:** 200, 204

### 16.1.1.1. Request

This table shows the header parameters for the show account details and list containers request:

Name	Type	Description
X-Auth-Token	String (Required)	Authentication token.

Name	Type	Description
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 <code>format</code> query parameter, set this header to <code>application/json</code> , <code>application/xml</code> , or <code>text/xml</code> .

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 in value than the specified marker.
end_marker	String (Optional)	For a string value $x$ , returns container names that are less in value than the specified marker.
format	String (Optional)	The response format. Valid values are <code>json</code> , <code>xml</code> , or <code>plain</code> . The default is <code>plain</code> .  If you append the <code>format=xml</code> or <code>format=json</code> query parameter to the storage account URL, the response shows extended container information serialized in the specified format.  If you append the <code>format=plain</code> 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.

### 16.1.1.2. Response

#### Example 16.1. 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 16.2. List containers response: HTTP and XML

```
HTTP/1.1 200 OK
Content-Length: 262
X-Account-Object-Count: 1
X-Timestamp: 1389453423.35964
X-Account-Meta-Subject: Literature
X-Account-Bytes-Used: 14
X-Account-Container-Count: 2
Content-Type: application/xml; charset=utf-8
Accept-Ranges: bytes
X-Trans-Id: tx69f60bc9f7634a01988e6-0052d9544b
Date: Fri, 17 Jan 2014 16:03:23 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<account name="my_account">
  <container>
    <name>janeausten</name>
    <count>0</count>
    <bytes>0</bytes>
  </container>
  <container>
    <name>marktwain</name>
    <count>1</count>
    <bytes>14</bytes>
  </container>
</account>
```

This operation does not return a response body.

## 16.1.2. 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 } 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 a metadata header, send an empty value for that particular 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 }: arbitrary value header. 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 204 status code.

To confirm your changes, issue a show account metadata request.

**Normal response codes:** 204

### 16.1.2.1. 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 an old and new key active at the same time.
X-Account-Meta-name	String <i>(Optional)</i>	The account metadata. The {name} is the name of metadata item that you want to add, update, or delete. To delete this item, send an empty value in this header.  You must specify a X-Account-Meta-{name} header for each metadata item (for each {name}) 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 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, update, or delete account metadata request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.

### 16.1.2.2. 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-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.

## 16.1.3. Show account metadata

Method	URI	Description
HEAD	/v1/{account}	Shows metadata for a specified 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.

This operation does not accept a request body.

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 401 Unauthorized error code.

**Normal response codes:** 204

**Error response codes:** unauthorized (401)

### 16.1.3.1. 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	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
	(Optional)	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.

### 16.1.3.2. Response

This table shows the header parameters for the show account metadata response:

Name	Type	Description
X-Account-Object-Count	Int (Required)	The number of objects in the account.
X-Account-Container-Count	Int (Required)	The number of containers.
X-Account-Bytes-Used	Int (Required)	The total number of bytes that are stored in Object Storage for the account.
X-Account-Meta-name	String (Optional)	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-Account-Meta-Temp-URL-Key	String (Optional)	The secret key value for temporary URLs. If not set, this header is not returned by this operation.
X-Account-Meta-Temp-URL-Key-2	String (Optional)	A second secret key value for temporary URLs. If not set, this header is not returned by this operation.
Content-Length	String (Required)	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 (Required)	If the operation fails, this value is the MIME type of the error text in the response body.
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.

## 16.2. Containers

List objects in a specified container. Create, show details for, and delete containers. Create, update, show, and delete container metadata.

Method	URI	Description
GET	/v1/{account}/{container}{?limit,marker,end_marker,prefix,format,delimiter,path}	Shows details for a specified container and lists objects, sorted by name, in the container.

Method	URI	Description
<b>PUT</b>	/v1/{account}/{container}	Creates a container.
<b>DELETE</b>	/v1/{account}/{container}	Deletes an empty container.
<b>POST</b>	/v1/{account}/{container}	Creates, updates, or deletes custom metadata for a container.
<b>HEAD</b>	/v1/{account}/{container}	Shows container metadata, including the number of objects and the total bytes of all objects stored in the container.

## 16.2.1. Show container details and list objects

Method	URI	Description
<b>GET</b>	/v1/{account}/{container}{?limit,marker,end_marker,prefix,format,delimiter,path}	Shows details for a specified 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:

- Show container details for and list objects in the `marktwain` container, and ask for a JSON response:

```
curl -i $publicURL/marktwain?format=json -X GET -H "X-Auth-Token: $token"
```

- Show container details for and list objects in the `marktwain` container, and ask for an XML response:

```
curl -i $publicURL/marktwain?format=xml -X GET -H "X-Auth-Token: $token"
```

If you use query parameters to page through a long list of objects, 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.

If the request succeeds, the operation returns one of these status codes:

- 200. Success. The response body lists the objects.
- 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 parameters, and you have reached the end of the list.

If the container does not exist, the 404 Not Found error code is returned.

**Normal response codes:** 200, 204

**Error response codes:** NotFound (404)

### 16.2.1.1. Request

This table shows the header parameters for the show container details and list objects request:

Name	Type	Description
X-Auth-Token	String	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 container details and list objects request:

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container.

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 $n$ , limits the number of results to $n$ .
marker	String (Optional)	For a string value $x$ , returns container names that are greater in value than the specified marker.
end_marker	String (Optional)	For a string value $x$ , returns container names that are less in value than the specified marker.
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 format=xml or format=json query parameter to the storage account URL, the response shows extended container information serialized in the specified format.  If you append the format=plain 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.

### 16.2.1.2. Response

#### Example 16.3. 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.

#### Example 16.4. Show container details response: HTTP and XML

```
HTTP/1.1 200 OK
Content-Length: 500
X-Container-Object-Count: 2
Accept-Ranges: bytes
X-Container-Meta-Book: TomSawyer
X-Timestamp: 1389727543.65372
X-Container-Bytes-Used: 26
Content-Type: application/xml; charset=utf-8
X-Trans-Id: txc75ea9a6e66f47d79e0c5-0052d6be76
Date: Wed, 15 Jan 2014 16:59:35 GMT
```

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

```
<object>
  <name>goodbye</name>
  <hash>451e372e48e0f6b1114fa0724aa79fa1</hash>
  <bytes>14</bytes>
  <content_type>application/octet-stream</content_type>
  <last_modified>2014-01-15T16:41:49.390270</last_modified>
</object>
<object>
  <name>helloworld</name>
  <hash>ed076287532e86365e841e92bfc50d8c</hash>
  <bytes>12</bytes>
  <content_type>application/octet-stream</content_type>
  <last_modified>2014-01-15T16:37:43.427570</last_modified>
</object>
</container>
```

This operation does not return a response body.

## 16.2.2. Create container

Method	URI	Description
PUT	/v1/{account}/{container}	Creates a container.

You do not need to check if 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

### 16.2.2.1. 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>	Sets an ACL that grants read access.
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	Enables versioning on this container. The value is the name of another container. You must UTF-8-encode and then URL-encode the name be-

Name	Type	Description
	(Optional)	fore you include it in the header. To disable versioning, set the header to an empty string.
X-Container-Meta-name	String (Optional)	The container metadata, where {name} is the name of metadata item.  You must specify a X-Container-Meta-{name} header for each metadata item (for each {name}) that you want to add or update.
Content-Type	String (Optional)	Changes the MIME type for the object.
X-Detect-Content-Type	Boolean (Optional)	If set to true, Object Storage guesses the content type based on the file extension and ignores the value sent in the Content-Type header, if present.
If-None-Match	String (Optional)	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.

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

Name	Type	Description
{account}	String	The unique name for the account. An account is also known as the project or tenant.
{container}	String	The unique name for the container.

### 16.2.2.2. Response

This table shows the header parameters for the create container response:

Name	Type	Description
Content-Length	String (Required)	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 (Required)	If the operation fails, this value is the MIME type of the error text in the response body.
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.

## 16.2.3. Delete container

Method	URI	Description
DELETE	/v1/{account}/{container}	Deletes an empty container.

This operation fails unless the container is empty. An empty container has no objects.

Delete the steven container:

```
curl -i $publicURL/steven -X DELETE -H "X-Auth-Token: $token"
```

If the container does not exist, the response is:

```
HTTP/1.1 404 Not Found
Content-Length: 70
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx4d728126b17b43b598bf7-0052d81e34
Date: Thu, 16 Jan 2014 18:00:20 GMT
```

If the container exists and the deletion succeeds, the response is:

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txf76c375ebece4df19c84c-0052d81f14
Date: Thu, 16 Jan 2014 18:04:04 GMT
```

If the container exists but is not empty, the response is:

```
HTTP/1.1 409 Conflict
Content-Length: 95
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx7782dc6a97b94a46956b5-0052d81f6b
Date: Thu, 16 Jan 2014 18:05:31 GMT

<html><h1>Conflict</h1><p>There was a conflict when trying to complete your
request.</p></html>
```

**Normal response codes:** 204

**Error response codes:** NotFound (404), Conflict (409)

### 16.2.3.1. Request

This table shows the header parameters for the delete container request:

Name	Type	Description
X-Auth-Token	String <i>(Required)</i>	Authentication token.

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.

Name	Type	Description
{container}	String	The unique name for the container.

### 16.2.3.2. 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-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.

## 16.2.4. Create, update, or delete container metadata

Method	URI	Description
<b>POST</b>	/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}: arbitrary value` header. 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.

This operation does not accept a request body.

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-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 204 status code.

To confirm your changes, issue a show container metadata request.

**Normal response codes:** 204

### 16.2.4.1. 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>	Sets an ACL that grants read access.
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.
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 a X-Container-Meta-{name} header for each metadata item (for each {name}) that you want to add or update.
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, 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.

Name	Type	Description
{container}	String	The unique name for the container.

## 16.2.4.2. Response

This table shows the header parameters for the create, update, or delete container meta-data 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-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.

## 16.2.5. Show container metadata

Method	URI	Description
<b>HEAD</b>	/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 204 status code.

**Normal response codes:** 204

### 16.2.5.1. Request

This table shows the header parameters for the show container metadata request:

Name	Type	Description
X-Auth-Token	String (Optional)	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
X-Newest	Boolean (Optional)	If set to 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 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.

### 16.2.5.2. Response

This table shows the header parameters for the show container metadata response:

Name	Type	Description
Content-Length	String (Required)	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
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-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.
Date	Datetime <i>(Required)</i>	The transaction date and time.

## 16.3. Objects

Create, replace, show details for, and delete objects. Copy objects from another object with a new or different name. Update object metadata.

Method	URI	Description
<b>GET</b>	/v1/{account}/{container}/{object}{?signature,expires,multipart-manifest}	Downloads the object content and gets the object metadata.
<b>PUT</b>	/v1/{account}/{container}/{object}{?multipart-manifest,signature,expires}	Creates a new object with specified data content and metadata, or replaces an existing object with specified data content and metadata.
<b>COPY</b>	/v1/{account}/{container}/{object}	Copies an object to another object in the object store.
<b>DELETE</b>	/v1/{account}/{container}/{object}{?multipart-manifest}	Permanently deletes an object from the object store.
<b>HEAD</b>	/v1/{account}/{container}/{object}{?signature,expires}	Shows object metadata.

Method	URI	Description
<b>POST</b>	/v1/{account}/{container}/{object}	Creates or updates object metadata.

## 16.3.1. Get object content and metadata

Method	URI	Description
GET	/v1/{account}/{container}/{object}{?signature,expires,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)

### 16.3.1.1. 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"> <li>• <b>Byte range specification.</b> 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.</li> <li>• <b>Suffix byte range specification.</b> Use LENGTH bytes to specify the length of the data range.</li> </ul> <p>The following forms of the header specify the following ranges of data:</p> <ul style="list-style-type: none"> <li>• Range: bytes=-5. The last five bytes.</li> <li>• Range: bytes=10-15. The five bytes of data after a 10-byte offset.</li> <li>• 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.</li> <li>• Range: bytes=4-6. Bytes 4 to 6 inclusive.</li> <li>• Range: bytes=2-2. Byte 2, the third byte of the data.</li> <li>• Range: bytes=6-. Byte 6 and after.</li> <li>• 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.</li> </ul>
If-Match	String <i>(Optional)</i>	See <a href="http://www.ietf.org/rfc/rfc2616.txt">http://www.ietf.org/rfc/rfc2616.txt</a> .
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 <a href="http://www.ietf.org/rfc/rfc2616.txt">http://www.ietf.org/rfc/rfc2616.txt</a> .
If-Unmodified-Since	Date <i>(Optional)</i>	See <a href="http://www.ietf.org/rfc/rfc2616.txt">http://www.ietf.org/rfc/rfc2616.txt</a> .

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.

Name	Type	Description
{container}	String	The unique name for the container.
{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
signature	String (Optional)	Used with temporary URLs to sign the request. For more information about temporary URLs, see <a href="#">OpenStack Object Storage API v1 Reference</a> .
expires	String (Optional)	Used with temporary URLs to specify the expiry time of the signature. For more information about temporary URLs, see <a href="#">OpenStack Object Storage API v1 Reference</a> .
multipart-manifest	String (Optional)	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.

### 16.3.1.2. Response

This table shows the header parameters for the get object content and metadata response:

Name	Type	Description
Content-Length	String (Required)	The length of the object content in the response body, in bytes.
Accept-Ranges	String (Required)	The type of ranges that the object accepts.
Last-Modified	String (Required)	The date and time that the object was created or the last time that the metadata was changed.
ETag	String (Required)	For objects smaller than 5 GB, this value is the MD5 checksum of the object content. The value is not quoted.  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.  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.
Content-Type	String (Required)	The MIME type of the object.
Content-Encoding	String (Optional)	If set, the value of the Content-Encoding metadata.  If not set, this header is not returned by this operation.
Content-Disposition	String (Optional)	If set, specifies the override behavior for the browser. For example, this header might specify that the browser use a download program to save this file rather than show the file, which is the default.  If not set, this header is not returned by this operation.
X-Delete-At	String (Optional)	If set, the time when the object will be deleted by the system in the format of a UNIX Epoch timestamp.  If not set, this header is not returned by this operation.

Name	Type	Description
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-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	Bool <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>	The transaction date and time.

## 16.3.2. Create or replace object

Method	URI	Description
<b>PUT</b>	/v1/{account}/{container}/{object}{?multipart-manifest,signature,expires}	Creates a new object with specified data content and metadata, or replaces an existing object with specified data content and metadata.

The **PUT** operation always creates a new 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 a 201 Created status 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 201 Created status code indicates a successful write.

If the request times out, the operation returns the 408 Request Timeout error code.

The 411 Length Required error 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 422 Unprocessable Entity error code.

**Normal response codes:** 201

**Error response codes:** timeout (408), lengthRequired (411), unprocessableEntity (422)

### 16.3.2.1. 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 <code>Content-Length</code> 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 <code>Content-Type</code> 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 <code>X-Copy-From</code> 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 <code>PUT</code> with <code>X-Copy-From</code> has the same effect as using the <code>COPY</code> 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 <code>Content-Encoding</code> metadata.
X-Delete-At	Int <i>(Optional)</i>	The certain date, in the format of a UNIX Epoch timestamp, when the object is removed.
X-Delete-After	Int <i>(Optional)</i>	Specifies the number of seconds after which the object is removed. Internally, the Object Storage system stores this value in the <code>X-Delete-At</code> metadata item.
X-Object-Meta-name	String <i>(Optional)</i>	The object metadata, where <code>{name}</code> is the name of the metadata item.  You must specify a <code>X-Object-Meta-{name}</code> header for each metadata 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.
{object}	String	The unique name for the object.

This table shows the query parameters for the create or replace object request:

Name	Type	Description
multipart-manifest	String <i>(Optional)</i>	If ?multipart-manifest=put, the object is a static large object manifest and the body contains the manifest.
signature	String <i>(Optional)</i>	Used with temporary URLs to sign the request. For more information about temporary URLs, see <a href="#">OpenStack Object Storage API v1 Reference</a> .
expires	String <i>(Optional)</i>	Used with temporary URLs to specify the expiry time of the signature. For more information about temporary URLs, see <a href="#">OpenStack Object Storage API v1 Reference</a> .

### 16.3.2.2. 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 ETag request header and the operation was successful, the values are the same.  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-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.

### 16.3.3. Copy object

Method	URI	Description
<b>COPY</b>	/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.

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 a new 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 a **201 Created** success node.

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

### 16.3.3.1. 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 a X-Object-Meta-{name} header for each metadata item (for each {name}) that you want to add or update.

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.
{object}	String	The unique name for the object.

### 16.3.3.2. 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.

Name	Type	Description
X-Copied-From-Last-Modified	String <i>(Optional)</i>	For a copied object, shows the last modified date and time for the container and object name from which the new object was copied.
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 form {container}/{object}.
Last-Modified	String <i>(Required)</i>	The date and time that the object was created or the last time that the metadata was changed.
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-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.

## 16.3.4. Delete object

Method	URI	Description
<b>DELETE</b>	/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 \$publi-cURL/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: tx36c7606fc1d1843f59167c-0052d6fdac
Date: Wed, 15 Jan 2014 21:29:16 GMT
```

Normally 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, ...

### 16.3.4.1. 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.
{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 <code>multipart-manifest=delete</code> query parameter and the object is a static large object, the segment objects and the manifest object are deleted. If you omit the <code>multipart-manifest=delete</code> query parameter and this is a static large object, the manifest object is deleted and 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 <b>DELETE</b> response has an empty body.

### 16.3.4.2. 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-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.

## 16.3.5. Show object metadata

Method	URI	Description
HEAD	/v1/{account}/{container}/{object}{?signature,expires}	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: tx37ea34dc1d48ca9bc7d-0052d84b6f
Date: Thu, 16 Jan 2014 21:13:19 GMT
```

If the request succeeds, the operation returns the 204 status code.

**Normal response codes:** 204

### 16.3.5.1. 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.
{object}	String	The unique name for the object.

This table shows the query parameters for the show object metadata request:

Name	Type	Description
signature	String <i>(Optional)</i>	Used with temporary URLs to sign the request. For more information about temporary URLs, see <a href="#">OpenStack Object Storage API v1 Reference</a> .
expires	String <i>(Optional)</i>	Used with temporary URLs to specify the expiry time of the signature. For more information about temporary URLs, see <a href="#">OpenStack Object Storage API v1 Reference</a> .

### 16.3.5.2. Response

This table shows the header parameters for the show object metadata response:

Name	Type	Description
Last-Modified	String <i>(Required)</i>	The date and time that the object was created or the last time that the metadata was changed.
Content-Length	String <i>(Required)</i>	The length of the object content in the response body, in bytes.
Content-Length	String <i>(Required)</i>	<b>HEAD</b> 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.
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.  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.  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.
Content-Encoding	String <i>(Optional)</i>	If set, the value of the Content-Encoding metadata.  If not set, this header is not returned by this operation.
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.  If not set, this header is not returned by this operation.
X-Delete-At	String <i>(Optional)</i>	If set, the time when the object will be deleted by the system in the format of a UNIX Epoch timestamp.  If not set, this header is not returned by this operation.
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-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-Static-Large-Object	Bool	Set to True if this object is a static large object manifest object.

Name	Type	Description
	(Required)	
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.

## 16.3.6. 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 also update these system metadata items: `Content-Type` `Content-Encoding` `Content-Disposition` `X-Delete-At`. 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. The form **POST** feature can also use the **POST** operation to upload objects. For more information about form **POST** see [OpenStack Object Storage API v1 Reference](#).

Example requests and responses:

- Create object metadata:

```
curl -i $publicURL/marktwain/goodbye -X POST -H "X-Auth-Token: $token" -H "X-Object-Meta-Book: GoodbyeColumbus"
```

```
HTTP/1.1 202 Accepted
Content-Length: 76
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txb5fb5c91ba1f4f37bb648-0052d84b3f
Date: Thu, 16 Jan 2014 21:12:31 GMT

<html><h1>Accepted</h1><p>The request is accepted for processing.</p></html>
```

- Update object metadata:

```
curl -i $publicURL/marktwain/goodbye -X POST -H "X-Auth-Token: $token" -H "X-Object-Meta-Book: GoodbyeOldFriend"
```

```

HTTP/1.1 202 Accepted
Content-Length: 76
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx5ec7ab81cdb34ced887c8-0052d84ca4
Date: Thu, 16 Jan 2014 21:18:28 GMT

<html><h1>Accepted</h1><p>The request is accepted for processing.</p></html>

```

**Normal response codes:** 202

### 16.3.6.1. 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 a 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 certain date, in the format of a UNIX Epoch timestamp, when the object is removed.
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>	Specifies the number of seconds after which the object is removed. 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.
{object}	String	The unique name for the object.

### 16.3.6.2. 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-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.

# 17. Orchestration API v1 (CURRENT)

Use a template language to orchestrate OpenStack services.

## 17.1. API versions

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

## 17.1.1. List versions

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

**Normal response codes:** 200

### 17.1.1.1. Request

This operation does not accept a request body.

### 17.1.1.2. Response

#### Example 17.1. List versions: JSON response

```
{
    "versions": [
        {
            "status": "CURRENT",
            "id": "v1.0",
            "links": [
                {
                    "href": "http://23.253.228.211:8000/v1/",
                    "rel": "self"
                }
            ]
        }
    ]
}
```

## 17.2. Stacks

Method	URI	Description
<b>POST</b>	/v1/{tenant_id}/stacks	Creates a stack.
<b>POST</b>	/v1/{tenant_id}/stacks	Creates a stack from existing resources.
<b>GET</b>	/v1/{tenant_id}/stacks{?status, name, limit, marker, sort_keys, sort_dir}	Lists active stacks.
<b>GET</b>	/v1/{tenant_id}/stacks/{stack_name}	Finds the canonical URL for a specified stack.
<b>GET</b>	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Shows details for a specified stack.
<b>PUT</b>	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Updates a specified stack.
<b>DELETE</b>	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Deletes a specified stack.
<b>POST</b>	/v1/{tenant_id}/stacks/preview	Previews a stack.
<b>DELETE</b>	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/abandon	Deletes a specified stack but leaves its resources intact, and returns data describing the stack and its resources.

## 17.2.1. Create stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks	Creates a stack.

**Normal response codes:** 201

### 17.2.1.1. Request

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

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

#### Example 17.2. Create stack: JSON request

```
{
    "stack_name": "{stack_name}",
    "template_url": "{template_url}",
    "parameters": {
        "param_name-1": "param_value-1",
        "param_name-2": "param_value-2"
    },
    "timeout_mins": "{timeout_mins}"
}
```

### 17.2.1.2. Response

#### Example 17.3. Create stack: JSON response

```
{
    "stack": {
        "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
        "links": [
            {
                "href": "http://192.168.123.200:8004/v1/
eb1c63a4f77141548385f113a28f0f52/stacks/simple_stack/3095aefc-09fb-4bc7-b1f0-
f21a304e864c",
                "rel": "self"
            }
        ]
    }
}
```

## 17.2.2. Adopt stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks	Creates a stack from existing resources.

**Normal response codes:** 201

### 17.2.2.1. Request

This table shows the URI parameters for the adopt stack request:

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

#### Example 17.4. Adopt stack: JSON request

```
{
    "stack_name": "{stack_name}",
    "template_url": "{template_url}",
    "timeout_mins": "{timeout_mins}",
    "adopt_stack_data": "{adopt_stack_data}"
}
```

### 17.2.2.2. Response

#### Example 17.5. Adopt stack: JSON response

```
{
    "stack": {
        "id": "5333af0c-cc26-47ee-ac3d-8784cefafbd7",
        "links": [
            {
                "href": "http://192.168.123.200:8004/v1/
eb1c63a4f77141548385f113a28f0f52/stacks/simple_stack/5333af0c-cc26-47ee-
ac3d-8784cefafbd7",
                "rel": "self"
            }
        ]
    }
}
```

## 17.2.3. List stack data

Method	URI	Description
GET	/v1/{tenant_id}/stacks{?status, name, limit, marker, sort_keys, sort_dir}	Lists active stacks.

Normal response codes: 200

### 17.2.3.1. Request

This table shows the URI parameters for the list stack data request:

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

This table shows the query parameters for the list stack data request:

Name	Type	Description
status	String <i>(Optional)</i>	Filters the stack list by the specified status. You can use this filter multiple times to filter by multiple statuses.
name	String <i>(Optional)</i>	Filters the stack list by the specified name.
limit	String <i>(Optional)</i>	Limits the number of stacks that appear on a page to this value. The typical pattern of limit and marker is to make an initial limited request and then to use the ID of the last stack from the response as the marker parameter in a subsequent limited request.
marker	String <i>(Optional)</i>	Specifies the ID of the last-seen stack. The typical pattern of limit and marker is to make an initial limited request and then to use the ID of the last stack from the response as the marker parameter in a subsequent limited request.
sort_keys	String <i>(Optional)</i>	Sorts the stack list by one of these attributes: name, status, created_at, or updated_at.
sort_dir	String <i>(Optional)</i>	The sort direction of the stack list. Either asc (ascending) or desc (descending).

This operation does not accept a request body.

### 17.2.3.2. Response

#### Example 17.6. List stack data: JSON response

```
{
  "stacks": [
    {
      "creation_time": "2014-06-03T20:59:46Z",
      "description": "sample stack",
      "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
      "links": [
        {
          "rel": "self",
          "href": "/v1/stacks/3095aefc-09fb-4bc7-b1f0-f21a304e864c"
        }
      ]
    }
  ]
}
```

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

## 17.2.4. Find stack

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}	Finds the canonical URL for a specified 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.

**Normal response codes:** 302

### 17.2.4.1. Request

This table shows the URI parameters for the find stack request:

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

This operation does not accept a request body.

## 17.2.5. Show stack details

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Shows details for a specified stack.

**Normal response codes:** 200

### 17.2.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.

This operation does not accept a request body.

### 17.2.5.2. Response

#### Example 17.7. 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::stack_id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
      "OS::stack_name": "simple_stack"
    },
    "stack_name": "simple_stack",
    "stack_status": "CREATE_COMPLETE",
    "stack_status_reason": "Stack CREATE completed successfully",
    "template_description": "sample stack",
    "timeout_mins": "",
    "updated_time": ""
  }
}
```

## 17.2.6. Update stack

Method	URI	Description
PUT	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Updates a specified stack.

**Normal response codes:** 202

### 17.2.6.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.

#### Example 17.8. Update stack: JSON request

```
{
    "template_url": "{template_url}",
    "parameters": {
        "param_name-1": "param_value-1",
        "param_name-2": "param_value-2"
    },
    "timeout_mins": "{timeout_mins}"
}
```

## 17.2.7. Delete stack

Method	URI	Description
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Deletes a specified stack.

**Normal response codes:** 204

### 17.2.7.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.

This operation does not accept a request body.

## 17.2.8. Preview stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks/preview	Previews a stack.

**Normal response codes:** 200

### 17.2.8.1. Request

This table shows the URI parameters for the preview stack request:

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

#### Example 17.9. Preview stack: JSON request

```
{
    "stack_name": "{stack_name}",
    "template_url": "{template_url}",
    "parameters": {
        "param_name-1": "param_value-1",
        "param_name-2": "param_value-2"
    },
    "timeout_mins": "{timeout_mins}"
}
```

### 17.2.8.2. Response

#### Example 17.10. Preview stack: JSON response

```
{
    "stack": {
        "id": "None",
        "stack_name": "sample_stack",
        "description": "Sample template description.",
        "template_description": "Sample template description.",
        "timeout_mins": 60,
        "disable_rollback": true,
        "capabilities": [],
        "notification_topics": [],
        "creation_time": "2014-02-19T16:04:56Z",
        "updated_time": "2014-02-19T16:04:56Z",
        "stack_status": "_",
        "stack_status_reason": "",
        "parameters": {
            "AWS::Region": "ap-southeast-1",
            "AWS::StackId": "arn:openstack:heat::2e327e5e7fa94b25a44be66fd9d1ec4d:stacks/sample_stack/None",
            "AWS::StackName": "sample_stack",
            "DBName": "wordpress",
            "DBPassword": "*****",
            "DBRootPassword": "*****",
            "DBUsername": "*****",
            "InstanceType": "m1.small",
            "KeyName": "heat_key",
            "LinuxDistribution": "F17"
        }
    }
}
```

```
        },
        "links": [
            {
                "href": "http://10.0.2.15:8004/v1/
2e327e5e7fa94b25a44be66fd9d1ec4d/stacks/sample_stack/None",
                "rel": "self"
            }
        ],
        "resources": [
            {
                "stack_name": "sample_stack",
                "resource_name": "WikiDatabase-1",
                "resource_type": "AWS::EC2::Instance",
                "description": "",
                "resource_action": "INIT",
                "resource_status": "COMPLETE",
                "resource_status_reason": "",
                "physical_resource_id": "",
                "required_by": [],
                "resource_identity": {
                    "path": "/resources/WikiDatabase-1",
                    "stack_id": "None",
                    "stack_name": "sample_stack",
                    "tenant": "2e327e5e7fa94b25a44be66fd9d1ec4d"
                },
                "stack_identity": {
                    "path": "",
                    "stack_id": "None",
                    "stack_name": "sample_stack",
                    "tenant": "2e327e5e7fa94b25a44be66fd9d1ec4d"
                },
                "updated_time": "2014-02-19T16:04:56Z",
                "metadata": {
                    "AWS::CloudFormation::Init": {
                        "config": {
                            "packages": {
                                "yum": {
                                    "httpd": [],
                                    "mysql": [],
                                    "mysql-server": [],
                                    "wordpress": []
                                }
                            }
                        },
                        "services": {
                            "systemd": {
                                "httpd": {
                                    "enabled": "true",
                                    "ensureRunning": "true"
                                },
                                "mysqld": {
                                    "enabled": "true",
                                    "ensureRunning": "true"
                                }
                            }
                        }
                    }
                }
            }
        ]
    }
}
```

```
        "other resources...": "other resources..."  
    }  
]  
}  
}
```

## 17.2.9. Abandon stack

Method	URI	Description
<b>DELETE</b>	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/abandon	Deletes a specified stack but leaves its resources intact, and returns data describing the stack and its resources.

**Normal response codes:** 200

### 17.2.9.1. Request

This table shows the URI parameters for the abandon stack request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.

This operation does not accept a request body.

### 17.2.9.2. Response

#### Example 17.11. Abandon stack: JSON response

```
{
    "status": "COMPLETE",
    "name": "g",
    "dry_run": true,
    "template": {
        "outputs": {
            "instance_ip": {
                "value": {
                    "str_replace": {
                        "params": {
                            "username": "ec2-user",
                            "hostname": {
                                "get_attr": [
                                    "server",
                                    "first_address"
                                ]
                            }
                        },
                        "template": "ssh username@hostname"
                    }
                }
            }
        }
    },
    "heat_template_version": "2013-05-23",
    "resources": {
        "server": {
            "type": "OS::Nova::Server",
            "properties": {
                "key_name": {
                    "get_param": "key_name"
                },
                "image": {

```

```

        "get_param": "image"
    },
    "flavor": {
        "get_param": "flavor"
    }
}
},
"parameters": {
    "key_name": {
        "default": "heat_key",
        "type": "string"
    },
    "image": {
        "default": "fedora-amd64",
        "type": "string"
    },
    "flavor": {
        "default": "m1.small",
        "type": "string"
    }
}
},
"action": "CREATE",
"id": "16934ca3-40e0-4fb2-a289-a700662ec05a",
"resources": {
    "server": {
        "status": "COMPLETE",
        "name": "server",
        "resource_data": {},
        "resource_id": "39d5dad7-7d7a-4cc8-bd84-851e9e2ff4ea",
        "action": "CREATE",
        "type": "OS::Nova::Server",
        "metadata": {}
    }
}
}

```

## 17.3. Stack actions

Performs non-lifecycle operations on the stack. Specify the action in the request body.

Method	URI	Description
<b>POST</b>	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Suspends a stack.
<b>POST</b>	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Resumes a suspended stack.

## 17.3.1. Suspend stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Suspends a stack.

Normal response codes: 201

### 17.3.1.1. Request

This table shows the URI parameters for the suspend stack request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.

#### Example 17.12. Suspend stack: JSON request

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

## 17.3.2. Resume stack

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Resumes a suspended stack.

Normal response codes: 201

### 17.3.2.1. Request

This table shows the URI parameters for the resume stack request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.

#### Example 17.13. Resume stack: JSON request

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

## 17.4. Stack resources

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/resources	Finds the canonical URL for the resource list of a specified stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources	Lists resources in a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}	Shows data for a specified resource.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/metadata	Shows metadata for a specified resource.
GET	/v1/{tenant_id}/resource_types	Lists the supported template resource types.
GET	/v1/{tenant_id}/resource_types/{type_name}	Shows the interface schema for a specified resource type.
GET	/v1/{tenant_id}/resource_types/{type_name}/template	Shows the template representation for a specified resource type.

## 17.4.1. Find stack resources

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/resources	Finds the canonical URL for the resource list of a specified stack.

**Normal response codes:** 302

### 17.4.1.1. Request

This table shows the URI parameters for the find stack resources request:

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

This operation does not accept a request body.

## 17.4.2. List resources

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources	Lists resources in a stack.

**Normal response codes:** 200

### 17.4.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.

This operation does not accept a request body.

## 17.4.3. Show resource data

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}	Shows data for a specified resource.

**Normal response codes:** 200

### 17.4.3.1. Request

This table shows the URI parameters for the show resource data request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.
{resource_name}	String	The name of a resource in the stack.

This operation does not accept a request body.

## 17.4.4. Show resource metadata

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/metadata	Shows metadata for a specified resource.

**Normal response codes:** 200

### 17.4.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.
{resource_name}	String	The name of a resource in the stack.

This operation does not accept a request body.

## 17.4.5. List resource types

Method	URI	Description
GET	/v1/{tenant_id}/resource_types	Lists the supported template resource types.

**Normal response codes:** 200

### 17.4.5.1. Request

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

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

This operation does not accept a request body.

## 17.4.6. Show resource schema

Method	URI	Description
GET	/v1/{tenant_id}/resource_types/{type_name}	Shows the interface schema for a specified resource type.

Normal response codes: 200

### 17.4.6.1. Request

This table shows the URI parameters for the show resource schema request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{type_name}	String	The name of a resource type.

This operation does not accept a request body.

### 17.4.6.2. Response

#### Example 17.14. Show resource schema: JSON response

```
{
    "attributes": {
        "an_attribute": {
            "description": "An attribute description ."
        }
    },
    "properties": {
        "a_property": {
            "update_allowed": false,
            "required": true,
            "type": "string",
            "description": "A resource description."
        }
    },
    "resource_type": "OS::Heat::AResourceName"
}
```

## 17.4.7. Show resource template

Method	URI	Description
GET	/v1/{tenant_id}/resource_types/{type_name}/template	Shows the template representation for a specified resource type.

Normal response codes: 200

### 17.4.7.1. Request

This table shows the URI parameters for the show resource template request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{type_name}	String	The name of a resource type.

This operation does not accept a request body.

## 17.5. Stack events

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/events	Finds the canonical URL for the event list of a specified stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/events	Lists events for a specified stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events	Lists events for a specified stack resource.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events/{event_id}	Shows details for a specified event.

## 17.5.1. Find stack events

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/events	Finds the canonical URL for the event list of a specified stack.

**Normal response codes:** 302

### 17.5.1.1. Request

This table shows the URI parameters for the find stack events request:

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

This operation does not accept a request body.

## 17.5.2. List stack events

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/events	Lists events for a specified stack.

**Normal response codes:** 200

### 17.5.2.1. Request

This table shows the URI parameters for the list stack events request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.

This operation does not accept a request body.

## 17.5.3. List resource events

Method	URI	Description
<b>GET</b>	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events	Lists events for a specified stack resource.

**Normal response codes:** 200

### 17.5.3.1. Request

This table shows the URI parameters for the list resource events request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.
{resource_name}	String	The name of a resource in the stack.

This operation does not accept a request body.

## 17.5.4. Show event details

Method	URI	Description
<b>GET</b>	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events/{event_id}	Shows details for a specified event.

**Normal response codes:** 200

### 17.5.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.
{resource_name}	String	The name of a resource in the stack.
{event_id}	String	The unique identifier of an event related to the resource in the stack.

This operation does not accept a request body.

## 17.6. Templates

Method	URI	Description
<b>GET</b>	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/template	Gets a template for a specified stack.
<b>POST</b>	/v1/{tenant_id}/validate	Validates a specified template.

## 17.6.1. Get stack template

Method	URI	Description
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/template	Gets a template for a specified stack.

**Normal response codes:** 200

### 17.6.1.1. Request

This table shows the URI parameters for the get stack template request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{stack_name}	String	The name of a stack.
{stack_id}	String	The unique identifier for a stack.

This operation does not accept a request body.

## 17.6.2. Validate template

Method	URI	Description
POST	/v1/{tenant_id}/validate	Validates a specified template.

Normal response codes: 200

### 17.6.2.1. Request

This table shows the URI parameters for the validate template request:

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

#### Example 17.15. Validate template: JSON request

```
{
    "template_url": "{template_url}"
}
```

## 17.7. Build info

Method	URI	Description
GET	/v1/{tenant_id}/build_info	Shows build information for a heat deployment.

## 17.7.1. Show build information

Method	URI	Description
GET	/v1/{tenant_id}/build_info	Shows build information for a heat deployment.

**Normal response codes:** 200

### 17.7.1.1. Request

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

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

This operation does not accept a request body.

### 17.7.1.2. Response

#### Example 17.16. Show build information: JSON response

```
{
    "api": {
        "revision": "{api_build_revision}"
    },
    "engine": {
        "revision": "{engine_build_revision}"
    }
}
```

## 17.8. 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 specified server.
GET	/v1/{tenant_id}/software_deployments/{deployment_id}	Shows details for a specified software deployment.
PUT	/v1/{tenant_id}/software_deployments/{deployment_id}	Updates a specified software deployment.
DELETE	/v1/{tenant_id}/software_deployments/{deployment_id}	Deletes a specified software deployment.



## 17.8.1. Create configuration

Method	URI	Description
POST	/v1/{tenant_id}/software_configs	Creates a software configuration.

**Normal response codes:** 200

### 17.8.1.1. Request

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

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

#### Example 17.17. 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
}
```

### 17.8.1.2. Response

#### Example 17.18. Create configuration: 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  
    }  
],  
"options": 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\" > $heat_outputs_path.result\necho \  
"Written to /tmp/$bar\"\nnecho \"Output to stderr\" 1>&2",  
    "id": "ddee7aca-aa32-4335-8265-d436b20db4f1"  
}  
}
```

## 17.8.2. Show configuration details

Method	URI	Description
GET	/v1/{tenant_id}/software_configs/{config_id}	Shows details for a software configuration.

**Normal response codes:** 200

### 17.8.2.1. Request

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

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

This operation does not accept a request body.

### 17.8.2.2. Response

#### Example 17.19. 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
            }
        ],
        "options": 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",
        "id": "ddee7aca-aa32-4335-8265-d436b20db4f1"
    }
}
```

}

## 17.8.3. Delete config

Method	URI	Description
DELETE	/v1/{tenant_id}/software_configs/{config_id}	Deletes a software configuration.

**Normal response codes:** 204

### 17.8.3.1. Request

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

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

This operation does not accept a request body.

## 17.8.4. List deployments

Method	URI	Description
GET	/v1/{tenant_id}/software_deployments	Lists all available software deployments.

**Normal response codes:** 200

### 17.8.4.1. Request

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

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

This operation does not accept a request body.

### 17.8.4.2. Response

#### Example 17.20. 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"
    }
  ]
}
```

## 17.8.5. Create deployment

Method	URI	Description
POST	/v1/{tenant_id}/software_deployments	Creates a software deployment.

**Normal response codes:** 200

### 17.8.5.1. Request

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

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

#### Example 17.21. Create deployment: JSON request

```
{
    "status": "IN_PROGRESS",
    "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
    "config_id": "8da95794-2ad9-4979-8ae5-739ce314c5cd",
    "stack_user_project_id": "c024bfada67845ddb17d2b0c0be8cd79",
    "action": "CREATE",
    "status_reason": "Deploy data available"
}
```

### 17.8.5.2. Response

#### Example 17.22. 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"
    }
}
```

## 17.8.6. Show server configuration metadata

Method	URI	Description
<b>GET</b>	/v1/{tenant_id}/software_deployments/metadata/{server_id}	Shows the deployment configuration metadata for a specified server.

Use the `group` property to specify the configuration hook to which the pass the metadata item.

**Normal response codes:** 200

### 17.8.6.1. Request

This table shows the URI parameters for the show server configuration metadata request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{server_id}	String	The ID of the server for which to fetch configuration metadata.

This operation does not accept a request body.

### 17.8.6.2. Response

#### Example 17.23. 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,
"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",
"id": "3d5ec2a8-7004-43b6-a7f6-542bdbbe9d434"
},
{
    "inputs": [
        {
            "default": null,
            "type": "String",
            "name": "foo",
            "value": "fu",
            "description": null
        },
        {
            "default": null,
            "type": "String",
            "name": "bar",
            "value": "barmy",
            "description": null
        }
    ]
}
```

```
        "type": "String",
        "name": "deploy_server_id",
        "value": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
        "description": "ID of the server being deployed to"
    },
    {
        "type": "String",
        "name": "deploy_action",
        "value": "CREATE",
        "description": "Name of the current action being deployed"
    },
    {
        "type": "String",
        "name": "deploy_stack_id",
        "value": "a/9bd57090-8954-48ab-bab9-adf9e1ac70fc",
        "description": "ID of the stack this deployment belongs
to"
    },
    {
        "type": "String",
        "name": "deploy_resource_name",
        "value": "other_deployment",
        "description": "Name of this deployment resource in the
stack"
    },
    {
        "type": "String",
        "name": "deploy_signal_id",
        "value": "http://192.168.20.103:8000/v1/signal/arn
%3Aopenstack%3Aheat%3A%3Ae2a84fbdaeb047ae8da4b503f3b69f1f%3Astacks%2Fa
%2F9bd57090-8954-48ab-bab9-adf9e1ac70fc%2Fresources%2Fother_deployment?
Timestamp=2014-03-19T20%3A30%3A59Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=
7b761482f8254946bcd3d5ccb36fe939&SignatureVersion=2&Signature=giMfv%2BhrAw6y
%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,
"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\"\\necho \"Output to stderr\" 1>&2",
"id": "8da95794-2ad9-4979-8ae5-739ce314c5cd"
}
]
```

## 17.8.7. Show deployment details

Method	URI	Description
GET	/v1/{tenant_id}/software_deployments/{deployment_id}	Shows details for a specified software deployment.

Normal response codes: 200

### 17.8.7.1. Request

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

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

This operation does not accept a request body.

### 17.8.7.2. Response

#### Example 17.24. 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"
  }
}
```

## 17.8.8. Update deployment

Method	URI	Description
PUT	/v1/{tenant_id}/software_deployments/{deployment_id}	Updates a specified software deployment.

**Normal response codes:** 200

### 17.8.8.1. Request

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

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

### Example 17.25. 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"
}
```

### 17.8.8.2. Response

### Example 17.26. 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"
    }
```

## 17.8.9. Delete deployment

Method	URI	Description
<b>DELETE</b>	/v1/{tenant_id}/software_deployments/{deployment_id}	Deletes a specified software deployment.

**Normal response codes:** 204

### 17.8.9.1. Request

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

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

This operation does not accept a request body.

# 18. Telemetry API v2 (CURRENT)

Manage telemetry operations.

## 18.1. Alarms

List, create, gets details for, update, and delete alarms.

Method	URI	Description
<b>GET</b>	/v2/alarms{?q}	Lists alarms, based on the specified query.
<b>POST</b>	/v2/alarms{?data}	Creates an alarm.
<b>GET</b>	/v2/alarms/{alarm_id}	Shows information for a specified alarm.
<b>PUT</b>	/v2/alarms/{alarm_id}{?data}	Updates a specified alarm.
<b>DELETE</b>	/v2/alarms/{alarm_id}	Deletes a specified alarm.
<b>PUT</b>	/v2/alarms/{alarm_id}/state{?state}	Sets the state of a specified alarm.
<b>GET</b>	/v2/alarms/{alarm_id}/state	Gets the state of a specified alarm.
<b>GET</b>	/v2/alarms/{alarm_id}/history{?q}	Assembles the history for a specified alarm.

## 18.1.1. List alarms

Method	URI	Description
GET	/v2/alarms{?q}	Lists alarms, based on the specified query.

**Normal response codes:** 200

### 18.1.1.1. Request

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

Name	Type	Description
q	List <i>(Optional)</i>	Filter rules for the alarms to be returned.

This operation does not accept a request body.

### 18.1.1.2. Response

#### Example 18.1. 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 18.2. 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.

## 18.1.2. Create alarm

Method	URI	Description
POST	/v2/alarms{?data}	Creates an alarm.

**Normal response codes:** 200

### 18.1.2.1. Request

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

Name	Type	Description
data	Alarm <i>(Optional)</i>	a alarm within the request body.

This operation does not accept a request body.

### 18.1.2.2. Response

#### Example 18.3. 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 18.4. 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.

## 18.1.3. Show alarm

Method	URI	Description
GET	/v2/alarms/{alarm_id}	Shows information for a specified alarm.

**Normal response codes:** 200

### 18.1.3.1. Request

This operation does not accept a request body.

### 18.1.3.2. Response

#### Example 18.5. Show 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 18.6. Show 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.

## 18.1.4. Update alarm

Method	URI	Description
PUT	/v2/alarms/{alarm_id}{?data}	Updates a specified alarm.

**Normal response codes:** 200

### 18.1.4.1. Request

This table shows the query parameters for the update alarm request:

Name	Type	Description
data	Alarm <i>(Optional)</i>	a alarm within the request body.

This operation does not accept a request body.

### 18.1.4.2. Response

#### Example 18.7. 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 18.8. 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.

## 18.1.5. Delete alarm

Method	URI	Description
<b>DELETE</b>	/v2/alarms/{alarm_id}	Deletes a specified alarm.

**Normal response codes:** 204

### 18.1.5.1. Request

This operation does not accept a request body.

## 18.1.6. Update alarm state

Method	URI	Description
PUT	/v2/alarms/{alarm_id}/state{?state}	Sets the state of a specified alarm.

**Normal response codes:** 200

### 18.1.6.1. Request

This table shows the query parameters for the update alarm state request:

Name	Type	Description
state	Dict <i>(Optional)</i>	An alarm state within the request body. Valid values are <code>ok</code> , <code>alarm</code> , or <code>insufficient data</code> .

This operation does not accept a request body.

## 18.1.7. Show alarm state

Method	URI	Description
GET	/v2/alarms/{alarm_id}/state	Gets the state of a specified alarm.

**Normal response codes:** 200

## 18.1.8. Show alarm history

Method	URI	Description
GET	/v2/alarms/{alarm_id}/history{?q}	Assembles the history for a specified alarm.

**Normal response codes:** 200

### 18.1.8.1. Request

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.

## 18.2. Meters

Get information for meters.

Method	URI	Description
GET	/v2/meters{?q}	Lists meters, based on the data recorded so far.
GET	/v2/meters/{meter_name}{?q,limit}	Gets samples for a specified meter.
POST	/v2/meters/{meter_name}{?samples}	Posts list of samples to Telemetry.
GET	/v2/meters/{meter_name}/statistics{?q,groupby,period}	Computes the statistics of the samples in a specified time range.

## 18.2.1. List meters

Method	URI	Description
GET	/v2/meters{?q}	Lists meters, based on the data recorded so far.

**Normal response codes:** 200

### 18.2.1.1. Request

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

Name	Type	Description
q	List <i>(Optional)</i>	Filter rules for the meters to be returned.

This operation does not accept a request body.

### 18.2.1.2. Response

#### Example 18.9. List meters: JSON response

```
[
  {
    "meter_id": "YmQ5NDMxYzEtOGQ2OS00YWQzLTgwM2EtOGQ0YTZiODlmZDM2K2luc3RhbmNl\n",
    "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 18.10. List meters: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
  <value>
    <meter_id>YmQ5NDMxYzEtOGQ2OS00YWQzLTgwM2EtOGQ0YTZiODlmZDM2K2luc3RhbmNl
    </meter_id>
    <name>instance</name>
    <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
    <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
    <source>openstack</source>
    <type>gauge</type>
    <unit>instance</unit>
    <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
  </value>
</values>
```

This operation does not return a response body.

## 18.2.2. Show meter

Method	URI	Description
GET	/v2/meters/{meter_name}{?q,limit}	Gets samples for a specified meter.

**Normal response codes:** 200

### 18.2.2.1. Request

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

Name	Type	Description
q	List <i>(Optional)</i>	Filter rules for the data to be returned.
limit	Int <i>(Optional)</i>	Maximum number of samples to return.

This operation does not accept a request body.

### 18.2.2.2. Response

#### Example 18.11. Show meter: JSON response

```
[
  {
    "counter_name": "instance",
    "counter_type": "gauge",
    "counter_unit": "instance",
    "counter_volume": 1.0,
    "message_id": "5460acce-4fd6-480d-ab18-9735ec7b1996",
    "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
    "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
    "resource_metadata": {
      "name1": "value1",
      "name2": "value2"
    },
    "source": "openstack",
    "timestamp": "2013-11-21T12:33:08.323533",
    "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff"
  }
]
```

#### Example 18.12. Show meter: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
  <value>
    <counter_name>instance</counter_name>
    <counter_type>gauge</counter_type>
    <counter_unit>instance</counter_unit>
    <counter_volume>1.0</counter_volume>
    <message_id>5460acce-4fd6-480d-ab18-9735ec7b1996</message_id>
    <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
```

```
<resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
<resource_metadata>
  <item>
    <key>name2</key>
    <value>value2</value>
  </item>
  <item>
    <key>name1</key>
    <value>value1</value>
  </item>
</resource_metadata>
<source>openstack</source>
<timestamp>2013-11-21T12:33:08.323533</timestamp>
<user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
</value>
</values>
```

This operation does not return a response body.

## 18.2.3. Create meter

Method	URI	Description
POST	/v2/meters/{meter_name}{?samples}	Posts list of samples to Telemetry.

**Normal response codes:** 200

### 18.2.3.1. Request

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

Name	Type	Description
samples	List <i>(Optional)</i>	A list of samples within the request body.

This operation does not accept a request body.

### 18.2.3.2. Response

#### Example 18.13. Create meter: JSON response

```
[  
  {  
    "counter_name": "instance",  
    "counter_type": "gauge",  
    "counter_unit": "instance",  
    "counter_volume": 1.0,  
    "message_id": "5460acce-4fd6-480d-ab18-9735ec7b1996",  
    "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",  
    "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",  
    "resource_metadata": {  
      "name1": "value1",  
      "name2": "value2"  
    },  
    "source": "openstack",  
    "timestamp": "2013-11-21T12:33:08.323533",  
    "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff"  
  }  
]
```

#### Example 18.14. Create meter: XML response

```
<?xml version="1.0" encoding="UTF-8"?>  
<values>  
  <value>  
    <counter_name>instance</counter_name>  
    <counter_type>gauge</counter_type>  
    <counter_unit>instance</counter_unit>  
    <counter_volume>1.0</counter_volume>  
    <message_id>5460acce-4fd6-480d-ab18-9735ec7b1996</message_id>  
    <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>  
    <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>  
    <resource_metadata>  
      <item>
```

```
<key>name2</key>
<value>value2</value>
</item>
<item>
  <key>name1</key>
  <value>value1</value>
</item>
</resource_metadata>
<source>openstack</source>
<timestramp>2013-11-21T12:33:08.323533</timestramp>
<user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
</value>
</values>
```

This operation does not return a response body.

## 18.2.4. Show meter statistics

Method	URI	Description
GET	/v2/meters/{meter_name}/statistics {?q,groupby,period}	Computes the statistics of the samples in a specified time range.

Normal response codes: 200

### 18.2.4.1. Request

This table shows the query parameters for the show meter statistics request:

Name	Type	Description
q	List <i>(Optional)</i>	Filter rules for the data to be returned.
groupby	List <i>(Optional)</i>	Fields for group by aggregation
period	Int <i>(Optional)</i>	Returned result will be an array of statistics for a period long of that number of seconds.

This operation does not accept a request body.

### 18.2.4.2. Response

#### Example 18.15. 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 18.16. 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.

## 18.3. Resources

Get information for resources.

Method	URI	Description
<b>GET</b>	/v2/resources{?q}	Lists definitions for all resources.
<b>GET</b>	/v2/resources/{resource_id}{?resource_id}	Gets details for a specified resource.

## 18.3.1. List resources

Method	URI	Description
GET	/v2/resources{?q}	Lists definitions for all resources.

**Normal response codes:** 200

### 18.3.1.1. Request

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

Name	Type	Description
q	List <i>(Optional)</i>	Filter rules for the resources to be returned.

This operation does not accept a request body.

### 18.3.1.2. Response

#### Example 18.17. 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",
    "timestamp": "2013-11-21T12:33:08.189843",
    "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff"
  }
]
```

#### Example 18.18. List resources: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
  <value>
    <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>
<metadata>
  <item>
    <key>name2</key>
    <value>value2</value>
  </item>
  <item>
    <key>name1</key>
    <value>value1</value>
  </item>
</metadata>
<project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
<resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
<source>openstack</source>
<timestampl>2013-11-21T12:33:08.189843</timestampl>
<user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
  </value>
</values>
```

This operation does not return a response body.

## 18.3.2. Show resource information

Method	URI	Description
GET	/v2/resources/{resource_id}{?resource_id}	Gets details for a specified resource.

Normal response codes: 200

### 18.3.2.1. Request

This table shows the query parameters for the show resource information request:

Name	Type	Description
resource_id	String <i>(Optional)</i>	The UUID of the resource.

This operation does not accept a request body.

### 18.3.2.2. Response

#### Example 18.19. Show resource information: 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",
  "timestamp": "2013-11-21T12:33:08.189843",
  "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff"
}
```

#### Example 18.20. Show resource information: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
<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>
  <metadata>
    <item>
      <key>name2</key>
      <value>value2</value>
    </item>
    <item>
      <key>name1</key>
      <value>value1</value>
    </item>
  </metadata>
  <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
  <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
  <source>openstack</source>
  <timestamp>2013-11-21T12:33:08.189843</timestamp>
  <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
</value>
```

This operation does not return a response body.