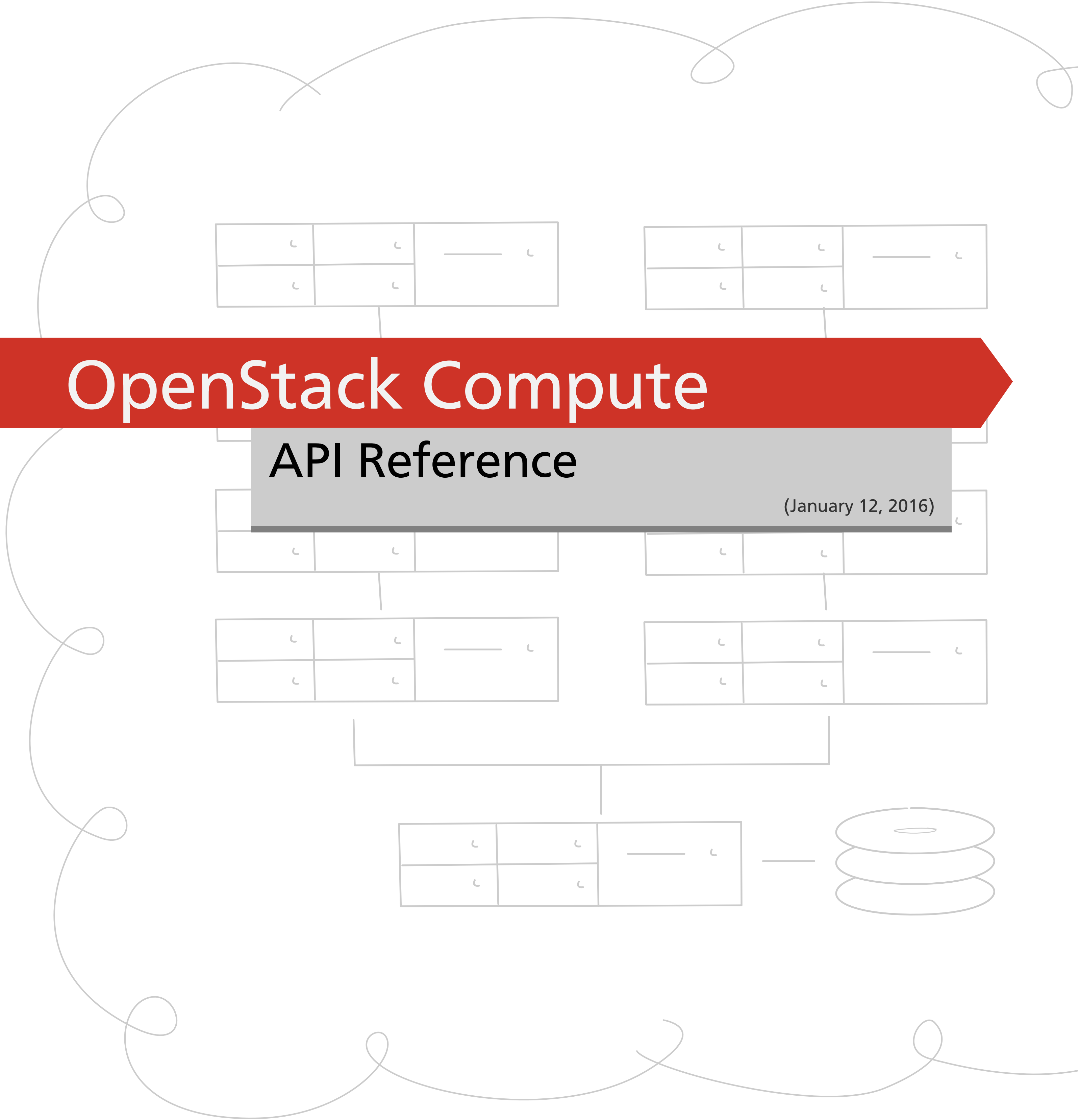

OpenStack Compute

API Reference

(January 12, 2016)

openstack™

docs.openstack.org

OpenStack Compute API Reference

Table of Contents

1. Compute API (CURRENT) 1
  - 1.1. API versions 10
    - 1.1.1. List API versions 11
  - 1.2. Servers (servers) 11
    - 1.2.1. List servers 14
    - 1.2.2. Create server 17
    - 1.2.3. List details for servers 20
    - 1.2.4. Show server details 23
    - 1.2.5. Update server 26
    - 1.2.6. Delete server 30
  - 1.3. Servers multiple create (servers) 31
    - 1.3.1. Create multiple servers 32
    - 1.3.2. Create multiple servers with reservation ID 34
  - 1.4. Servers actions (servers, action) 35
    - 1.4.1. Add (associate) fixed IP (addFixedIp action) 38
    - 1.4.2. Add (associate) floating IP (addFloatingIp action) 39
    - 1.4.3. Attach volume (attach action) 40
    - 1.4.4. Confirm resized server (confirmResize action) 41
    - 1.4.5. Create image (createImage action) 42
    - 1.4.6. Evacuate server (evacuate action) 44
    - 1.4.7. Force-delete server (forceDelete action) 45
    - 1.4.8. Lock server (lock action) 46
    - 1.4.9. Pause server (pause action) 47
    - 1.4.10. Show console output (os-getConsoleOutput action) 48
    - 1.4.11. Get RDP console (os-getRDPConsole action) 49
    - 1.4.12. Get serial console (os-getSerialConsole action) 50
    - 1.4.13. Get SPICE console (os-getSPICEConsole action) 51
    - 1.4.14. Get VNC console (os-getVNCConsole action) 52
    - 1.4.15. Reboot server (reboot action) 53
    - 1.4.16. Rebuild server (rebuild action) 54
    - 1.4.17. Remove (disassociate) fixed IP (removeFixedIp action) 57
    - 1.4.18. Remove (disassociate) floating IP (removeFloatingIp action) 58
    - 1.4.19. Rescue server (rescue action) 59
    - 1.4.20. Resize server (resize action) 60
    - 1.4.21. Restore soft-deleted instance (restore action) 61
    - 1.4.22. Resume suspended server (resume action) 62
    - 1.4.23. Revert resized server (revertResize action) 63
    - 1.4.24. Shelve server (shelve action) 64
    - 1.4.25. Shelf-offload (remove) server (shelveOffload action) 66
    - 1.4.26. Start server (os-start action) 67
    - 1.4.27. Stop server (os-stop action) 68
    - 1.4.28. Unlock server (unlock action) 69
    - 1.4.29. Unpause server (unpause action) 70
    - 1.4.30. Unrescue server (unrescue action) 71
    - 1.4.31. Unshelve (restore) shelved server (unshelve action) 72
  - 1.5. Servers admin actions (servers, action) 72
    - 1.5.1. Create server back up (createBackup action) 74
    - 1.5.2. Inject network information (injectNetworkInfo action) 75

1.5.3. Migrate server (migrate action) 76
1.5.4. Live-migrate server (os-migrateLive action) 77
1.5.5. Reset networking on a server (resetNetwork action) 78
1.5.6. Reset server state (os-resetState action) 79
1.6. Servers diagnostics (servers, diagnostics) 79
1.6.1. Show server diagnostics 80
1.7. Servers IPs (servers, ips) 80
1.7.1. List IPs 82
1.7.2. Show IP details 83
1.8. Server metadata (servers, metadata) 83
1.8.1. List all metadata 85
1.8.2. Create or replace metadata items 86
1.8.3. Update metadata items 87
1.8.4. Show metadata item details 88
1.8.5. Create or update metadata item 89
1.8.6. Delete metadata item 90
1.9. Servers action (servers, os-instance-actions) 90
1.9.1. List actions for server 91
1.9.2. Show server action details 92
1.10. Port interfaces (servers, os-interface) 93
1.10.1. Create interface 94
1.10.2. List port interfaces 95
1.10.3. Show port interface details 96
1.10.4. Detach interface 97
1.11. Servers password (servers, os-server-password) 97
1.11.1. Show server password 98
1.11.2. Clear admin password 99
1.12. Servers virtual interfaces (servers, os-virtual-interfaces) 99
1.12.1. List virtual interfaces 100
1.13. Flavors with extended attributes (flavors) 100
1.13.1. List flavors 101
1.13.2. Show flavor details 103
1.13.3. List flavors with details 104
1.14. Flavors access (flavors, os-flavor-access, action) 106
1.14.1. List flavor access information for given flavor 107
1.14.2. Add flavor access to tenant 108
1.14.3. Remove flavor access from tenant 109
1.15. Flavors extra-specs (flavors, os-flavor-extra-specs) 109
1.15.1. Create extra specs for a flavor 111
1.15.2. List extra specs for a flavor 112
1.15.3. Show an extra spec for a flavor 113
1.15.4. Update an extra spec for a flavor 114
1.15.5. Delete an extra spec for a flavor 115
1.16. Flavors manage (flavors, os-flavor-manage) 115
1.16.1. Create flavor 116
1.16.2. Delete flavor 117
1.17. Keypairs (keypairs) 117
1.17.1. List keypairs 118
1.17.2. Create or import keypair 119
1.17.3. Delete keypair 120
1.17.4. Show keypair details 121

1.18. Limits (limits) ... 121
1.18.1. Show rate and absolute limits ... 122
1.19. Extensions (extensions) ... 122
1.19.1. List extensions ... 124
1.19.2. Show extension details ... 138
1.20. Images ... 138
1.20.1. List images ... 140
1.20.2. List images details ... 145
1.20.3. Show image details ... 151
1.20.4. Delete image ... 153
1.21. Image metadata ... 153
1.21.1. Show image metadata ... 154
1.21.2. Create or replace image metadata ... 155
1.21.3. Update image metadata items ... 156
1.21.4. Show image metadata item details ... 157
1.21.5. Create or update image metadata item ... 158
1.21.6. Delete image metadata item ... 159
1.22. Guest agents (os-agents) ... 159
1.22.1. Create agent build ... 160
1.22.2. List agent builds ... 161
1.22.3. Delete agent build ... 162
1.22.4. Update agent build ... 163
1.23. Host aggregates (os-aggregates, action) ... 163
1.23.1. Create aggregate ... 165
1.23.2. List aggregates ... 166
1.23.3. Show aggregate details ... 167
1.23.4. Update aggregate ... 168
1.23.5. Add host ... 169
1.23.6. Create or update aggregate metadata ... 170
1.23.7. Remove host ... 171
1.24. Assisted volume snapshots (os-assisted-volume-snapshots) ... 171
1.24.1. Create assisted volume snapshots ... 173
1.24.2. Delete assisted volume snapshot ... 174
1.25. Availability zones (os-availability-zone) ... 174
1.25.1. Get availability zone information ... 175
1.25.2. Get detailed availability zone information ... 176
1.26. Bare metal nodes (os-baremetal-nodes) ... 177
1.26.1. Add bare metal node ... 178
1.26.2. List bare metal nodes ... 180
1.26.3. Add interface to bare metal node ... 181
1.26.4. Delete interface from bare metal node ... 182
1.26.5. Show bare metal node details ... 183
1.26.6. Delete bare metal node ... 185
1.27. Servers with block device mapping format (servers, os-block-device-mapping) ... 185
1.27.1. Create server with block device mapping ... 186
1.28. Cells (os-cells, capacities) ... 187
1.28.1. List cells ... 188
1.28.2. List cells with details ... 190
1.28.3. Show cell data ... 191
1.28.4. Show cell capacities ... 192

1.29. Root certificates (os-certificates) 193
1.29.1. Create certificate 194
1.29.2. Show certificate details 196
1.30. Cloudpipe (os-cloudpipe) 196
1.30.1. List cloudpipes 198
1.30.2. Create cloudpipe 199
1.30.3. Update cloudpipe 200
1.31. Server consoles (servers, os-consoles, os-console-auth-token) 200
1.31.1. Create console 201
1.31.2. Lists consoles 202
1.31.3. Show console details 203
1.31.4. Delete console 204
1.31.5. Show console authentication token 205
1.32. Fixed IPs (os-fixed-ips) 205
1.32.1. Show fixed IP details 206
1.32.2. Reserve or release a fixed IP 207
1.33. Floating IP DNS records (os-floating-ip-dns) 207
1.33.1. List DNS domains 208
1.33.2. Create or update DNS domain 209
1.33.3. Delete DNS domain 210
1.33.4. Create or update DNS entry 211
1.33.5. Find unique DNS entry 212
1.33.6. Delete DNS entry 213
1.33.7. List DNS entries 214
1.34. Floating IP pools (os-floating-ip-pools) 214
1.34.1. List floating IP pools 215
1.35. Floating IPs (os-floating-ips) 215
1.35.1. List floating IP addresses 217
1.35.2. Create (allocate) floating IP address 218
1.35.3. Show floating IP address details 219
1.35.4. Delete (deallocate) floating IP address 220
1.36. Floating IPs bulk (os-floating-ips-bulk) 220
1.36.1. List floating IPs 221
1.36.2. Create floating IPs 222
1.36.3. Bulk-delete floating IPs 223
1.36.4. List floating IPs by host 224
1.37. Ping instances (os-fping) 224
1.37.1. Ping instances 225
1.37.2. Ping an instance 227
1.38. Hosts (os-hosts) 227
1.38.1. List hosts 228
1.38.2. Enable host 230
1.38.3. Show host details 231
1.38.4. Reboot host 232
1.38.5. Shut down host 233
1.38.6. Start host 234
1.39. Hypervisors (os-hypervisors) 234
1.39.1. List hypervisors 235
1.39.2. Show hypervisor statistics 236
1.39.3. Show hypervisor details 237
1.39.4. Show hypervisor uptime 239

1.40. Instance usage audit log (os-instance-usage-audit-log) 239
1.40.1. List usage audits for an instance 240
1.40.2. List usage audits before specified time 241
1.41. Migrations (os-migrations) 242
1.41.1. List migrations 243
1.42. Networks (os-networks) 244
1.42.1. Create network 245
1.42.2. List networks 247
1.42.3. Add network 249
1.42.4. Show network details 250
1.42.5. Delete network 252
1.42.6. Associate host 253
1.42.7. Disassociate host 254
1.42.8. Disassociate network 255
1.42.9. Disassociate project 256
1.43. Quota class (os-quota-class-sets) 256
1.43.1. Show quota 257
1.43.2. Update quota 258
1.44. Quota sets (os-quota-sets) 259
1.44.1. Delete quota for tenant 260
1.44.2. Force-update quota 261
1.44.3. Update quota 262
1.44.4. Show default quotas 263
1.44.5. Show quotas (detailed) 264
1.44.6. Update quota for user 265
1.44.7. Delete quota for user 266
1.45. Security groups (os-security-groups) 266
1.45.1. List security groups 267
1.45.2. Create security group 268
1.45.3. Show security group details 269
1.45.4. Update security group 270
1.45.5. Delete security group 271
1.45.6. List security groups by server 272
1.46. Rules for default security group (os-security-group-default-rules) 272
1.46.1. List default security group rules 273
1.46.2. Create default security group rule 274
1.46.3. Show default security group rule details 275
1.46.4. Delete default security group rule 276
1.47. Rules for security group (os-security-group-rules) 276
1.47.1. Create security group rule 277
1.47.2. Delete security group rule 278
1.48. Create external events (os-server-external-events) 278
1.48.1. Run events 279
1.49. Server groups (os-server-groups) 279
1.49.1. List server groups 281
1.49.2. Create server group 282
1.49.3. Show server group details 283
1.49.4. Delete server group 284
1.50. Usage reports (os-simple-tenant-usage) 284
1.50.1. List tenant usage for all tenants 285
1.50.2. Show usage details for tenant 286

1.51. Project networks (os-tenant-networks) 286
1.51.1. Create project network 288
1.51.2. List project networks 289
1.51.3. Show project network details 291
1.51.4. Delete project network 292
1.51.5. Associate host 293
1.51.6. Disassociate host 294
1.51.7. Disassociate network 295
1.51.8. Disassociate project 296
1.52. Volume extension (os-volumes, os-snapshots) 296
1.52.1. List volumes 298
1.52.2. Create volume 299
1.52.3. List volumes with details 300
1.52.4. Show volume details 301
1.52.5. Delete volume 302
1.52.6. List volume types 303
1.52.7. Show volume type details 304
1.52.8. Create snapshot 305
1.52.9. List snapshots 306
1.52.10. List snapshots with details 307
1.52.11. Show snapshot details 309
1.52.12. Delete snapshot 310

1. Compute API (CURRENT)

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

Each API version lists the extensions that it requires to make the full request and responses available. API v2.1 must enable all extensions all the time. It uses micro-version headers to expose any additional functionality.

Note

Effective in the OpenStack Kilo release, the Compute API no longer supports XML in requests and responses.

For details about the Compute API, see the Compute API.

Method	URI	Description
API versions		
GET	/	Lists information about all Compute API versions.
Servers (servers)		
GET	/v2.1/{tenant_id}/servers{?changes-since,image,flavor,name,status,host,limit,marker}	Lists IDs, names, and links for all servers.
POST	/v2.1/{tenant_id}/servers	Creates a server.
GET	/v2.1/{tenant_id}/servers/detail{?changes-since,image,flavor,name,status,host,limit,marker}	Lists all servers with details.
GET	/v2.1/{tenant_id}/servers/{server_id}	Shows details for a server.
PUT	/v2.1/{tenant_id}/servers/{server_id}	Updates the editable attributes of a server.
DELETE	/v2.1/{tenant_id}/servers/{server_id}	Deletes a server.
Servers multiple create (servers)		
POST	/v2.1/{tenant_id}/servers	Creates one or more servers.
POST	/v2.1/{tenant_id}/servers	Creates one or more servers with a reservation ID.
Servers actions (servers, action)		
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a fixed IP address to a server instance, which associates that address with the server. The fixed IP address is retrieved from the network that you specify in the request.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to a server, which associates that address with the server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Attaches a volume to a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Confirms a pending resize action for a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Creates an image from a server.
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Evacuates a server from a failed host to a new one.

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Force-deletes a server before deferred cleanup.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Locks a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Pauses a server. Changes its status to `PAUSED`.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Shows console output for a server instance.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Gets an RDP console for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Gets a serial console for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Gets a SPICE console for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Gets a VNC console for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Reboots a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Rebuilds a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Removes, or disassociates, a fixed IP address from a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Removes, or disassociates, a floating IP address from a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Puts a server in rescue mode and changes its status to `RESCUE`.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resizes a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Restores a previously soft-deleted server instance. You cannot use this method to restore deleted instances.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resumes a suspended server and changes its status to `ACTIVE`.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Cancels and reverts a pending resize action for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Shelves a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Shelf-offloads, or removes, a shelved server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Starts a stopped server and changes its status to `ACTIVE`.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Stops a running server and changes its status to `SHUTOFF`.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Unlocks a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Unpauses a paused server and changes its status to `ACTIVE`.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Unrescues a server. Changes status to `ACTIVE`.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Unshelves, or restores, a shelved server.
Servers admin actions (servers, action)		
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Creates a back up of a server.

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Injects network information into a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Migrates a server to a host. The scheduler chooses the host.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Live-migrates a server to a new host without rebooting.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resets networking on a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resets the state of a server.
	Servers diagnostics (servers, diagnostics)	
GET	`/v2.1/{tenant_id}/servers/{server_id}/diagnostics`	Shows basic usage data for a server.
	Servers IPs (servers, ips)	
GET	`/v2.1/{tenant_id}/servers/{server_id}/ips`	Lists IP addresses that are assigned to an instance.
GET	`/v2.1/{tenant_id}/servers/{server_id}/ips/{network_label}`	Shows IP addresses details for a network label of a server instance.
	Server metadata (servers, metadata)	
GET	`/v2.1/{tenant_id}/servers/{server_id}/metadata`	Lists all metadata for a server.
PUT	`/v2.1/{tenant_id}/servers/{server_id}/metadata`	Creates or replaces one or more metadata items for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/metadata`	Updates one or more metadata items for a server.
GET	`/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}`	Shows details for a metadata item, by key, for a server.
PUT	`/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}`	Creates or replaces a metadata item, by key, for a server.
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}`	Deletes a metadata item, by key, from a server.
	Servers action (servers, os-instance-actions)	
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-instance-actions`	Lists actions for a server.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-instance-actions/{request_id}`	Shows details for an action and server.
	Port interfaces (servers, os-interface)	
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-interface`	Creates a port interface and uses it to attach a port to a server instance.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-interface`	Lists port interfaces that are attached to a server.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}`	Shows details for a port interface that is attached to a server.
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}`	Detaches a port interface.
	Servers password (servers, os-server-password)	
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-server-password`	Shows the administrative password for a server.

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/os-server-password`	Clears the encrypted administrative password for a server, which removes it from the metadata server.
Servers virtual interfaces (servers, os-virtual-interfaces)		
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-virtual-interfaces`	Lists the virtual interfaces for an instance.
Flavors with extended attributes (flavors)		
GET	`/v2.1/{tenant_id}/flavors`	Lists flavors.
GET	`/v2.1/{tenant_id}/flavors/{flavor_id}`	Shows details for a flavor.
GET	`/v2.1/{tenant_id}/flavors/detail`	Lists flavors with details.
Flavors access (flavors, os-flavor-access, action)		
GET	`/v2.1/{tenant_id}/flavors/os-flavor-access`	Lists flavor access information.
POST	`/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action`	Adds flavor access to a tenant and flavor.
POST	`/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action`	Removes flavor access from a tenant and flavor.
Flavors extra-specs (flavors, os-flavor-extra-specs)		
POST	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs`	Creates extra specs for a flavor, by ID.
GET	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs`	Lists all extra specs for a flavor, by ID.
GET	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}`	Shows an extra spec, by key, for a flavor, by ID.
PUT	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}`	Updates an extra spec, by key, for a flavor, by ID.
DELETE	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}`	Deletes an extra spec, by key, for a flavor, by ID.
Flavors manage (flavors, os-flavor-manage)		
POST	`/v2.1/{tenant_id}/flavors/os-flavor-manage`	Creates a flavor.
DELETE	`/v2.1/{tenant_id}/flavors/os-flavor-manage/{flavor_id}`	Deletes a flavor.
Keypairs (keypairs)		
GET	`/v2.1/{tenant_id}/os-keypairs`	Lists keypairs that are associated with the account.
POST	`/v2.1/{tenant_id}/os-keypairs`	Generates or imports a keypair.
DELETE	`/v2.1/{tenant_id}/os-keypairs/{keypair_name}`	Deletes a keypair.
GET	`/v2.1/{tenant_id}/os-keypairs/{keypair_name}`	Shows details for a keypair that is associated with the account.
Limits (limits)		
GET	`/v2.1/{tenant_id}/limits`	Shows rate and absolute limits for the tenant.
Extensions (extensions)		
GET	`/v2.1/{tenant_id}/extensions`	Lists available extensions.
GET	`/v2.1/{tenant_id}/extensions/{alias}`	Shows details for an extension, by alias.
Images		

Method	URI	Description
GET	`/v2.1/{tenant_id}/images{?changes-since,server,name,status,minDisk, minRam,type,limit,marker}`	Lists IDs, names, and links for available images.
GET	`/v2.1/{tenant_id}/images/detail{?changes-since,server,name,status, minDisk,minRam,type,limit,marker}`	Lists all details for available images.
GET	`/v2.1/{tenant_id}/images/{image_id}`	Shows details for an image.
DELETE	`/v2.1/{tenant_id}/images/{image_id}`	Deletes an image.
	Image metadata	
GET	`/v2.1/{tenant_id}/images/{image_id}/metadata`	Shows metadata for an image.
PUT	`/v2.1/{tenant_id}/images/{image_id}/metadata`	Creates or replaces metadata for an image.
POST	`/v2.1/{tenant_id}/images/{image_id}/metadata`	Updates metadata items, by key, for an image.
GET	`/v2.1/{tenant_id}/images/{image_id}/metadata/{key}`	Shows details for a metadata item, by key, for an image.
PUT	`/v2.1/{tenant_id}/images/{image_id}/metadata/{key}`	Creates or updates a metadata item, by key, for an image.
DELETE	`/v2.1/{tenant_id}/images/{image_id}/metadata/{key}`	Deletes a metadata item, by key, for an image.
	Guest agents (os-agents)	
POST	`/v2.1/{tenant_id}/os-agents`	Creates an agent build.
GET	`/v2.1/{tenant_id}/os-agents`	Lists agent builds.
DELETE	`/v2.1/{tenant_id}/os-agents`	Deletes an existing agent build.
PUT	`/v2.1/{tenant_id}/os-agents/{id}`	Updates an agent build.
	Host aggregates (os-aggregates, action)	
POST	`/v2.1/{tenant_id}/os-aggregates`	Creates an aggregate in an availability zone.
GET	`/v2.1/{tenant_id}/os-aggregates`	Lists all aggregates. Includes the ID, name, and availability zone for each aggregate.
GET	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}`	Shows the details of an aggregate, including hosts and metadata.
PUT	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}`	Updates either or both the name and availability zone for an aggregate.
POST	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action`	Adds a host to an aggregate.
POST	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action`	Creates or replaces metadata for an aggregate.
POST	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action`	Removes a host from an aggregate.
	Assisted volume snapshots (os-assisted-volume-snapshots)	
POST	`/v2.1/{tenant_id}/os-assisted-volume-snapshots`	Creates an assisted volume snapshot.
DELETE	`/v2.1/{tenant_id}/os-assisted-volume-snapshots/{snapshot_id}{?delete_info}`	Deletes an assisted volume snapshot.
	Availability zones (os-availability-zone)	
GET	`/v2.1/{tenant_id}/os-availability-zone`	Gets availability zone information.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-availability-zone/detail`	Gets detailed availability zone information.
Bare metal nodes (os-baremetal-nodes)		
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes`	Adds a bare metal node to a server.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes`	Lists the bare metal nodes that are associated with a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action`	Adds an interface to a bare metal node that is associated with a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action`	Deletes an interface from a bare metal node that is associated with a server.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}`	Shows details for a bare metal node.
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}`	Deletes a bare metal node from a server.
Servers with block device mapping format (servers, os-block-device-mapping)		
POST	`/v2.1/{tenant_id}/servers`	Creates a server with a block device mapping.
Cells (os-cells, capacities)		
GET	`/v2.1/{tenant_id}/os-cells`	Lists cells.
GET	`/v2.1/{tenant_id}/os-cells`	Lists cells with details.
GET	`/v2.1/{tenant_id}/os-cells/{cell_id}`	Shows data for a cell.
GET	`/v2.1/{tenant_id}/os-cells/{cell_id}/capacities`	Shows capacities for a cell.
Root certificates (os-certificates)		
POST	`/v2.1/{tenant_id}/os-certificates`	Creates a certificate.
GET	`/v2.1/{tenant_id}/os-certificates/{certificate_id}`	Shows details for a certificate.
Cloudpipe (os-cloudpipe)		
GET	`/v2.1/{tenant_id}/os-cloudpipe`	Lists cloudpipes.
POST	`/v2.1/{tenant_id}/os-cloudpipe`	Creates a cloudpipe.
POST	`/v2.1/{tenant_id}/os-cloudpipe/configure-project`	Updates the virtual private network (VPN) IP address and port for a cloudpipe instance.
Server consoles (servers, os-consoles, os-console-auth-token)		
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-consoles`	Creates a console for a server instance.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-consoles`	Lists all consoles for a server instance.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-consoles/{console_id}`	Shows details for a console for a server instance.
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/os-consoles/{console_id}`	Deletes a console for a server instance.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-console-auth-token`	Shows the authentication token for a console for a server instance.
Fixed IPs (os-fixed-ips)		

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}`	Shows details for a fixed IP address.
POST	`/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}/action`	Reserves or releases a fixed IP.
Floating IP DNS records (os-floating-ip-dns)		
GET	`/v2.1/{tenant_id}/os-floating-ip-dns`	Lists registered DNS domains published by the DNS drivers.
PUT	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}`	Creates or updates a DNS domain.
DELETE	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}`	Deletes a DNS domain and all associated host entries.
PUT	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}`	Creates or updates a DNS entry.
GET	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}`	Finds a unique DNS entry for a domain and name.
DELETE	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}`	Deletes a DNS entry.
GET	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}`	Lists DNS entries for a domain and IP.
Floating IP pools (os-floating-ip-pools)		
GET	`/v2.1/{tenant_id}/os-floating-ip-pools`	Lists floating IP pools.
Floating IPs (os-floating-ips)		
GET	`/v2.1/{tenant_id}/os-floating-ips`	Lists floating IP addresses associated with the tenant or account.
POST	`/v2.1/{tenant_id}/os-floating-ips`	Creates, or allocates, a floating IP address for the current project. By default, the floating IP address is allocated from the public pool.
GET	`/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}`	Shows details for a floating IP address, by ID, that is associated with the tenant or account.
DELETE	`/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}`	Deletes, or deallocates, a floating IP address from the current project and returns it to the pool from which it was allocated.
Floating IPs bulk (os-floating-ips-bulk)		
GET	`/v2.1/{tenant_id}/os-floating-ips-bulk`	Lists all floating IPs.
POST	`/v2.1/{tenant_id}/os-floating-ips-bulk`	Bulk-creates floating IPs.
POST	`/v2.1/{tenant_id}/os-floating-ips-bulk/delete`	Bulk-deletes floating IPs.
GET	`/v2.1/{tenant_id}/os-floating-ips-bulk/{host_name}`	Lists all floating IPs for a host.
Ping instances (os-fping)		
GET	`/v2.1/{tenant_id}/os-fping{?all_tenants,include,exclude}`	Run the fping utility to ping instances and report which ones are alive.
GET	`/v2.1/{tenant_id}/os-fping/{id}`	Run the fping utility to ping an instance and report whether it is alive.
Hosts (os-hosts)		
GET	`/v2.1/{tenant_id}/os-hosts`	Lists hosts.
PUT	`/v2.1/{tenant_id}/os-hosts/{host_name}`	Enables or puts a host in maintenance mode.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-hosts/{host_name}`	Shows details for a host.
GET	`/v2.1/{tenant_id}/os-hosts/{host_name}/reboot`	Reboots a host.
GET	`/v2.1/{tenant_id}/os-hosts/{host_name}/shutdown`	Shuts down a host.
GET	`/v2.1/{tenant_id}/os-hosts/{host_name}/startup`	Starts a host.
Hypervisors (os-hypervisors)		
GET	`/v2.1/{tenant_id}/os-hypervisors`	Lists hypervisors.
GET	`/v2.1/{tenant_id}/os-hypervisors/statistics`	Shows summary statistics for all hypervisors over all compute nodes.
GET	`/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}`	Shows details for a hypervisor.
GET	`/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}/uptime`	Shows the uptime for a hypervisor.
Instance usage audit log (os-instance-usage-audit-log)		
GET	`/v2.1/{tenant_id}/os-instance_usage_audit_log`	Lists usage audits for an instance.
GET	`/v2.1/{tenant_id}/os-instance_usage_audit_log/{before_timestamp}{?before_timestamp}`	Lists usage audits that occurred before a specified time.
Migrations (os-migrations)		
GET	`/v2.1/{tenant_id}/os-migrations`	Lists in-progress migrations.
Networks (os-networks)		
POST	`/v2.1/{tenant_id}/os-networks`	Creates a network.
GET	`/v2.1/{tenant_id}/os-networks`	Lists networks for the project.
POST	`/v2.1/{tenant_id}/os-networks/add`	Adds a network to a project.
GET	`/v2.1/{tenant_id}/os-networks/{id}`	Shows details for a network.
DELETE	`/v2.1/{tenant_id}/os-networks/{id}`	Deletes a network.
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Associates a network with a host.
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Disassociates a host from a network.
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Disassociates a network from a project. You can then reuse the network.
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Disassociates a project from a network.
Quota class (os-quota-class-sets)		
GET	`/v2.1/{tenant_id}/os-quota-class-sets/{class_id}`	Shows the quota for a class.
PUT	`/v2.1/{tenant_id}/os-quota-class-sets/{class_id}`	Updates quota for a class.
Quota sets (os-quota-sets)		
DELETE	`/v2.1/{tenant_id}/os-quota-sets`	Deletes a quota for tenant.
PUT	`/v2.1/{tenant_id}/os-quota-sets`	Force-updates quota for tenant.
PUT	`/v2.1/{tenant_id}/os-quota-sets`	Updates quota for tenant.
GET	`/v2.1/{tenant_id}/os-quota-sets/defaults`	Shows default quotas for tenant.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-quota-sets/detail`	Lists quotas with details for a tenant.
PUT	`/v2.1/{tenant_id}/os-quota-sets/{?user_id}`	Updates quota for user.
DELETE	`/v2.1/{tenant_id}/os-quota-sets/{?user_id}`	Deletes quota for a user.
Security groups (os-security-groups)		
GET	`/v2.1/{tenant_id}/os-security-groups`	Lists security groups.
POST	`/v2.1/{tenant_id}/os-security-groups`	Creates a security group.
GET	`/v2.1/{tenant_id}/os-security-groups/{security_group_id}`	Shows details for a security group.
PUT	`/v2.1/{tenant_id}/os-security-groups/{security_group_id}`	Updates a security group.
DELETE	`/v2.1/{tenant_id}/os-security-groups/{security_group_id}`	Deletes a security group.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-security-groups`	Lists security groups for a server.
Rules for default security group (os-security-group-default-rules)		
GET	`/v2.1/{tenant_id}/os-security-group-default-rules`	Lists default security group rules.
POST	`/v2.1/{tenant_id}/os-security-group-default-rules`	Creates a default security group rule.
GET	`/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}`	Shows details for a security group rule.
DELETE	`/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}`	Deletes a security group rule.
Rules for security group (os-security-group-rules)		
POST	`/v2.1/{tenant_id}/os-security-group-rules`	Creates a rule for a security group.
DELETE	`/v2.1/{tenant_id}/os-security-group-rules/{security_group_rule_id}`	Deletes a security group rule.
Create external events (os-server-external-events)		
POST	`/v2.1/{tenant_id}/os-server-external-events`	Creates one or more external events, which the API dispatches to the instance.
Server groups (os-server-groups)		
GET	`/v2.1/{tenant_id}/os-server-groups{?all_projects}`	Lists all server groups for the tenant.
POST	`/v2.1/{tenant_id}/os-server-groups`	Creates a server group.
GET	`/v2.1/{tenant_id}/os-server-groups/{server_group_id}`	Shows details for a server group.
DELETE	`/v2.1/{tenant_id}/os-server-groups/{server_group_id}`	Deletes a server group.
Usage reports (os-simple-tenant-usage)		
GET	`/v2.1/os-simple-tenant-usage`	Lists usage information for all tenants.
GET	`/v2.1/os-simple-tenant-usage/{tenant_id}`	Shows usage details for a tenant.
Project networks (os-tenant-networks)		

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-tenant-networks`	Creates a project network.
GET	`/v2.1/{tenant_id}/os-tenant-networks`	Lists all project networks.
GET	`/v2.1/{tenant_id}/os-tenant-networks/{id}`	Shows details for a project network.
DELETE	`/v2.1/{tenant_id}/os-tenant-networks/{id}`	Deletes a project network.
POST	`/v2.1/{tenant_id}/os-tenant-networks/{id}/action`	Associates a network with a host.
POST	`/v2.1/{tenant_id}/os-tenant-networks/{id}/action`	Disassociates a host from a network.
POST	`/v2.1/{tenant_id}/os-tenant-networks/{id}/action`	Disassociates a network from a project so that the network can be reused.
POST	`/v2.1/{tenant_id}/os-tenant-networks/{id}/action`	Disassociates a project from a network.
Volume extension (os-volumes, os-snapshots)		
GET	`/v2.1/{tenant_id}/os-volumes`	Lists the volumes associated with the account.
POST	`/v2.1/{tenant_id}/os-volumes`	Creates a volume.
GET	`/v2.1/{tenant_id}/os-volumes/detail`	Lists all volumes with details.
GET	`/v2.1/{tenant_id}/os-volumes/{volume_id}`	Shows details for a volume.
DELETE	`/v2.1/{tenant_id}/os-volumes/{volume_id}`	Deletes a volume.
GET	`/v2.1/{tenant_id}/os-volume-types`	Lists volume types.
GET	`/v2.1/{tenant_id}/os-volume-types/{volume_type_id}`	Shows details for a volume type.
POST	`/v2.1/{tenant_id}/os-snapshots`	Creates a snapshot.
GET	`/v2.1/{tenant_id}/os-snapshots`	Lists snapshots.
GET	`/v2.1/{tenant_id}/os-snapshots/detail`	Lists all snapshots with details.
GET	`/v2.1/{tenant_id}/os-snapshots/{snapshot_id}`	Shows details for a snapshot.
DELETE	`/v2.1/{tenant_id}/os-snapshots/{snapshot_id}`	Deletes a snapshot from the account.

1.1. API versions

Lists information for all API versions.

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

1.1.1. List API versions

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

Normal response codes: 200, 300

1.1.1.1. Request

This operation does not accept a request body.

1.1.1.2. Response

Example 1.1. List API versions: JSON response

```
{
    "versions": [
        {
            "id": "v2.0",
            "links": [
                {
                    "href": "http://openstack.example.com/v2/",
                    "rel": "self"
                }
            ],
            "status": "SUPPORTED",
            "version": "",
            "min_version": "",
            "updated": "2011-01-21T11:33:21Z"
        },
        {
            "id": "v2.1",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/",
                    "rel": "self"
                }
            ],
            "status": "CURRENT",
            "version": "2.14",
            "min_version": "2.1",
            "updated": "2013-07-23T11:33:21Z"
        }
    ]
}
```

1.2. Servers (servers)

Lists, creates, shows details for, updates, and deletes servers.

Passwords

When you create a server, you can specify a password through the optional adminPass attribute. The password must meet the complexity requirements set by your OpenStack Compute provider. The server might enter an `ERROR` state if the complexity requirements are not met. In this case, a client might issue a change password action to reset the server password.

If you do not specify a password, the API generates and assigns a random password that it returns in the response object. This password meets the security requirements set by the compute provider. For security reasons, subsequent **GET** calls do not require this password.

Server metadata

You can specify custom server metadata at server launch time. The maximum size for each metadata key-value pair is 255 bytes. The compute provider determines the maximum number of key-value pairs for each server. You can query this value through the `maxServerMeta` absolute limit.

Server networks

You can specify one or more networks to which the server connects at launch time. Users can also specify a specific port on the network or the fixed IP address to assign to the server interface.

> **Note**
>
> You can use both IPv4 and IPv6 addresses as access addresses and you can assign both addresses simultaneously. You can update access addresses after you create a server.

Server personality

To customize the personality of a server instance, you can inject data into its file system. For example, you might insert ssh keys, set configuration files, or store data that you want to retrieve from inside the instance. This customization method provides minimal launch-time personalization. If you require significant customization, create a custom image.

Follow these guidelines when you inject files:

- The maximum size of the file path data is 255 bytes.

- Encode the file contents as a Base64 string. The compute provider determines the maximum size of the file contents. The image that you use to create the server determines this value.

  > **Note**
  >
  > The maximum limit refers to the number of bytes in the decoded data and not to the number of characters in the encoded data.

- The `maxPersonality` absolute limit defines the maximum number of file path and content pairs that you can supply. The compute provider determines this value.

- The `maxPersonalitySize` absolute limit is a byte limit that applies to all images in the deployment. Providers can set additional per-image personality limits.

The file injection might not occur until after the server builds and boots.

After file injection, only system administrators can access personality files. For example, on Linux, all files have root as the owner and the root group as the group owner, and allow only user and group read access (`chmod 440`).

Server access addresses

In a hybrid environment, the underlying implementation might not control the IP address of a server. Instead, the access IP address might be part of the dedicated hardware; for example, a router/NAT device. In this case, you cannot use the addresses that the implementation provides to access the server from outside the local LAN. Instead, the API might assign a separate *access address* at creation time to provide access to the server. This address might not be directly bound to a network interface on the server and might not necessarily appear when you query the server addresses. However, clients should use an access address to access the server directly.

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers{?changes-since,image,flavor,name,status,host,limit,marker}`	Lists IDs, names, and links for all servers.
POST	`/v2.1/{tenant_id}/servers`	Creates a server.
GET	`/v2.1/{tenant_id}/servers/detail{?changes-since,image,flavor,name,status,host,limit,marker}`	Lists all servers with details.
GET	`/v2.1/{tenant_id}/servers/{server_id}`	Shows details for a server.
PUT	`/v2.1/{tenant_id}/servers/{server_id}`	Updates the editable attributes of a server.
DELETE	`/v2.1/{tenant_id}/servers/{server_id}`	Deletes a server.

1.2.1. List servers

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers{?changes-since,image,flavor,name,status,host,limit,marker}`	Lists IDs, names, and links for all servers.

Servers contain a status attribute that indicates the current server state. You can filter on the server status when you complete a list servers request. The server status is returned in the response body. The possible server status values are:

Server status values

- `ACTIVE`. The server is active.
- `BUILDING`. The server has not finished the original build process.
- `DELETED`. The server is permanently deleted.
- `ERROR`. The server is in error.
- `HARD_REBOOT`. The server is hard rebooting. This is equivalent to pulling the power plug on a physical server, plugging it back in, and rebooting it.
- `MIGRATING`. The server is being migrated to a new host.
- `PASSWORD`. The password is being reset on the server.
- `PAUSED`. In a paused state, the state of the server is stored in RAM. A paused server continues to run in frozen state.
- `REBOOT`. The server is in a soft reboot state. A reboot command was passed to the operating system.
- `REBUILD`. The server is currently being rebuilt from an image.
- `RESCUED`. The server is in rescue mode. A rescue image is running with the original server image attached.
- `RESIZED`. Server is performing the differential copy of data that changed during its initial copy. Server is down for this stage.
- `REVERT_RESIZE`. The resize or migration of a server failed for some reason. The destination server is being cleaned up and the original source server is restarting.
- `SOFT_DELETED`. The server is marked as deleted but the disk images are still available to restore.
- `STOPPED`. The server is powered off and the disk image still persists.
- `SUSPENDED`. The server is suspended, either by request or necessity. This status appears for only the XenServer/XCP, KVM, and ESXi hypervisors. Administrative users can suspend an instance if it is infrequently used or to perform system maintenance. When you

suspend an instance, its VM state is stored on disk, all memory is written to disk, and the virtual machine is stopped. Suspending an instance is similar to placing a device in hibernation; memory and vCPUs become available to create other instances.

- `UNKNOWN`. The state of the server is unknown. Contact your cloud provider.
- `VERIFY_RESIZE`. System is awaiting confirmation that the server is operational after a move or resize.

Normal response codes: 200

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

1.2.1.1. Request

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

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

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

Name	Type	Description
`changes-since`	DateTime *(Optional)*	Filters the response by a date and time stamp when the server last changed status.
`image`	UUID *(Optional)*	Filters the response by an image, as a UUID.
`flavor`	UUID *(Optional)*	Filters the response by a flavor, as a UUID. A flavor is a combination of memory, disk size, and CPUs.
`name`	Regexp *(Optional)*	Filters the response by a server name, as a string. You can use regular expressions in the query. For example, the ?`name=bob` regular expression returns both bob and bobb. If you must match on only bob, you can use a regular expression that matches the syntax of the underlying database server that is implemented for Compute, such as MySQL or PostgreSQL.
`status`	ServerStatus *(Optional)*	Filters the response by a server status, as a string. For example, `ACTIVE`.
`host`	String *(Optional)*	Filters the response by a host name, as a string. This query parameter is typically available to only administrative users. If you are a non-administrative user, the API ignores this parameter.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.2.1.2. Response

Example 1.2. List servers: JSON response

```
{
    "servers": [
        {
            "id": "a291599e-6de2-41a6-88df-c443ddcef70d",
            "links": [
                {
                    "href": "http://openstack.example.com/v2/openstack/
servers/a291599e-6de2-41a6-88df-c443ddcef70d",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/servers/
a291599e-6de2-41a6-88df-c443ddcef70d",
                    "rel": "bookmark"
                }
            ],
            "name": "new-server-test"
        }
    ]
}
```

1.2.2. Create server

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers`	Creates a server.

The progress of this operation depends on the location of the requested image, network I/O, host load, selected flavor, and other factors.

To check the progress of the request, make a `GET /servers/{id}` request. This call returns a progress attribute, which is a percentage value from 0 to 100.

The `Location` header returns the full URL to the newly created server and is available as a `self` and `bookmark` link in the server representation.

Note

When you create a server, the response shows only the server ID, its links, and the admin password. You can get additional attributes through subsequent **GET** requests on the server.

Include the `block-device-mapping-v2` parameter in the create request body to boot a server from a volume.

Include the `key_name` parameter in the create request body to add a keypair to the server when you create it. To create a keypair, make a create keypair request.

Preconditions

- The user must have sufficient server quota to create the number of servers requested.
- The connection to the Image service is valid.

Asynchronous postconditions

- With correct permissions, you can see the server status as `ACTIVE` through API calls.
- With correct access, you can see the created server in the compute node that OpenStack Compute manages.

Troubleshooting

- If the server status remains `BUILDING` or shows another error status, the request failed. Ensure you meet the preconditions then investigate the compute node.
- The server is not created in the compute node that OpenStack Compute manages.
- The compute node needs enough free resource to match the resource of the server creation request.
- Ensure that the scheduler selection filter can fulfill the request with the available compute nodes that match the selection criteria of the filter.

Normal response codes: 202

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

1.2.2.1. Request

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

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

This list shows the body parameters for the request:

- **parameters**:

  - **block_device_mapping_v2**:

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

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

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

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

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

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

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

      To delete the boot volume when the server is destroyed, specify `true`. Otherwise, specify `false`.

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

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

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

      Defines the order in which a hypervisor tries devices when it attempts to boot the guest from storage.

      Give each device a unique boot index starting from `0`. To disable a device from booting, set the boot index to a negative value or use the default boot index value, which is `None`.

      The simplest usage is, set the boot index of the boot device to `0` and use the default boot index value, `None`, for any other devices.

      Some hypervisors might not support booting from multiple devices; these hypervisors consider only the device with a boot index of `0`.

Some hypervisors support booting from multiple devices but only if the devices are of different types. For example, a disk and CD-ROM.

Example 1.3. Create server: JSON request

```
{
    "server": {
        "name": "new-server-test",
        "imageRef": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "flavorRef": "http://openstack.example.com/flavors/1",
        "metadata": {
            "My Server Name": "Apache1"
        }
    }
}
```

1.2.2.2. Response

Example 1.4. Create server: JSON response

```
{
    "server": {
        "OS-DCF:diskConfig": "AUTO",
        "adminPass": "zPnp2GseTqG4",
        "id": "8195065c-fea4-4d57-b93f-5c5c63fe90e8",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/servers/
8195065c-fea4-4d57-b93f-5c5c63fe90e8",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
8195065c-fea4-4d57-b93f-5c5c63fe90e8",
                "rel": "bookmark"
            }
        ],
        "security_groups": [
            {
                "name": "default"
            }
        ]
    }
}
```

1.2.3. List details for servers

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

The compute provisioning algorithm has an anti-affinity property that attempts to spread customer VMs across hosts. Under certain situations, VMs from the same customer might be placed on the same host. The hostId property shows the host that your server runs on and can be used to determine this scenario if it is relevant to your application.

For each server, shows server details including configuration drive, extended status, and server usage information.

The extended status information appears in the `OS-EXT-STS:vm_state`, `OS-EXT-STS:power_state`, and `OS-EXT-STS:task_state` attributes.

The server usage information appears in the `OS-SRV-USG:launched_at` and `OS-SRV-USG:terminated_at` attributes.

To hide `addresses` information for instances in a certain state, set the `osapi_hide_server_address_states` configuration option. Set this option to a valid VM state in the `nova.conf` configuration file.

Note

HostId is unique *per account* and is not globally unique.

Normal response codes: 200

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

1.2.3.1. Request

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

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

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

Name	Type	Description
`changes-since`	DateTime *(Optional)*	Filters the response by a date and time stamp when the server last changed status.
`image`	UUID *(Optional)*	Filters the response by an image, as a UUID.
`flavor`	UUID *(Optional)*	Filters the response by a flavor, as a UUID. A flavor is a combination of memory, disk size, and CPUs.

Name	Type	Description
`name`	Regexp *(Optional)*	Filters the response by a server name, as a string. You can use regular expressions in the query. For example, the ?`name=bob` regular expression returns both bob and bobb. If you must match on only bob, you can use a regular expression that matches the syntax of the underlying database server that is implemented for Compute, such as MySQL or PostgreSQL.
`status`	ServerStatus *(Optional)*	Filters the response by a server status, as a string. For example, ACTIVE.
`host`	String *(Optional)*	Filters the response by a host name, as a string. This query parameter is typically available to only administrative users. If you are a non-administrative user, the API ignores this parameter.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.2.3.2. Response

Example 1.5. List details for servers: JSON response

```
{
    "servers": [
        {
            "addresses": {
                "private": [
                    {
                        "addr": "192.168.0.3",
                        "OS-EXT-IPS-MAC:mac_addr": "aa:bb:cc:dd:ee:ff",
                        "OS-EXT-IPS:type": "fixed",
                        "version": 4
                    }
                ]
            },
            "created": "2013-09-23T13:53:12Z",
            "flavor": {
                "id": "1",
                "links": [
                    {
                        "href": "http://openstack.example.com/openstack/
flavors/1",
                        "rel": "bookmark"
                    }
                ]
            },
            "hostId":
 "f1e160ad2bf07084f3d3e0dfdd0795d80da18a60825322c15775c0dd",
            "id": "9cbefc35-d372-40c5-88e2-9fda1b6ea12c",
            "image": {
                "id": "70a599e0-31e7-49b7-b260-868f441e862b",
                "links": [
                    {
                        "href": "http://openstack.example.com/openstack/
images/70a599e0-31e7-49b7-b260-868f441e862b",
                        "rel": "bookmark"
                    }
                ]
            },
            "key_name": null,
            "links": [
                {
                    "href": "http://openstack.example.com/v2/openstack/
servers/9cbefc35-d372-40c5-88e2-9fda1b6ea12c",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/servers/
9cbefc35-d372-40c5-88e2-9fda1b6ea12c",
                    "rel": "bookmark"
                }
            ],
            "metadata": {
                "My Server Name": "Apache1"
            },
            "name": "new-server-test",
            "accessIPv4": "",
            "accessIPv6": "",
            "config_drive": "",
            "OS-DCF:diskConfig": "AUTO",
            "OS-EXT-AZ:availability_zone": "nova",
            "OS-EXT-SRV-ATTR:host": "c3f14e9812ad496baf92ccfb3c61e15f",
            "OS-EXT-SRV-ATTR:hypervisor_hostname": "fake-mini",
            "OS-EXT-SRV-ATTR:instance_name": "instance-00000001",
            "OS-EXT-STS:power_state": 1,
            "OS-EXT-STS:task_state": null,
            "OS-EXT-STS:vm_state": "active",
            "os-extended-volumes:volumes_attached": [],
            "OS-SRV-USG:launched_at": "2013-09-23T13:53:12.774549",
            "OS-SRV-USG:terminated_at": null,
            "progress": 0,
            "security_groups": [
                {
                    "name": "default"
                }
            ],
            "status": "ACTIVE",
            "tenant_id": "openstack",
            "updated": "2013-10-31T06:32:32Z",
            "user_id": "fake"
        }
    ]
}
```

1.2.4. Show server details

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}`	Shows details for a server.

Includes server details including configuration drive, extended status, and server usage information.

The extended status information appears in the `OS-EXT-STS:vm_state`, `OS-EXT-STS:power_state`, and `OS-EXT-STS:task_state` attributes.

The server usage information appears in the `OS-SRV-USG:launched_at` and `OS-SRV-USG:terminated_at` attributes.

To hide `addresses` information for instances in a certain state, set the `osapi_hide_server_address_states` configuration option. Set this option to a valid VM state in the `nova.conf` configuration file.

Note

HostId is unique *per account* and is not globally unique.

Preconditions

- The server must exist.

Normal response codes: 200

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

1.2.4.1. Request

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

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

This operation does not accept a request body.

1.2.4.2. Response

Example 1.6. Show server details: JSON response

```
{
    "server": {
        "addresses": {
            "private": [
                {
                    "addr": "192.168.0.3",
                    "OS-EXT-IPS-MAC:mac_addr": "aa:bb:cc:dd:ee:ff",
                    "OS-EXT-IPS:type": "fixed",
                    "version": 4
                }
            ]
        },
        "created": "2013-09-23T13:37:00Z",
        "flavor": {
            "id": "1",
            "links": [
                {
                    "href": "http://openstack.example.com/openstack/flavors/
1",
                    "rel": "bookmark"
                }
            ]
        },
        "hostId": "9cc36101a27c2a69c1a18241f6228454d9d7f466bd90c62db8e8b856",
        "id": "f474386b-4fb6-4e1f-b1d5-d6bf4437f7d5",
        "image": {
            "id": "70a599e0-31e7-49b7-b260-868f441e862b",
            "links": [
                {
                    "href": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                    "rel": "bookmark"
                }
            ]
        },
        "key_name": null,
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/servers/
f474386b-4fb6-4e1f-b1d5-d6bf4437f7d5",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
f474386b-4fb6-4e1f-b1d5-d6bf4437f7d5",
                "rel": "bookmark"
            }
        ],
        "metadata": {
            "My Server Name": "Apache1"
        },
        "name": "new-server-test",
        "accessIPv4": "192.0.2.0",
        "accessIPv6": "2002:0:0:0:0:0:c000:20e",
        "config_drive": "",
        "OS-DCF:diskConfig": "AUTO",
        "OS-EXT-AZ:availability_zone": "nova",
        "OS-EXT-SRV-ATTR:host": "b8b357f7100d4391828f2177c922ef93",
        "OS-EXT-SRV-ATTR:hypervisor_hostname": "fake-mini",
        "OS-EXT-SRV-ATTR:instance_name": "instance-00000001",
        "OS-EXT-STS:power_state": 1,
        "OS-EXT-STS:task_state": null,
        "OS-EXT-STS:vm_state": "active",
        "os-extended-volumes:volumes_attached": [],
        "OS-SRV-USG:launched_at": "2013-09-23T13:37:00.880302",
        "OS-SRV-USG:terminated_at": null,
        "progress": 0,
        "security_groups": [
            {
                "name": "default"
            }
        ],
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2013-10-31T07:31:30Z",
        "user_id": "fake"
    }
}
```

1.2.5. Update server

Method	URI	Description
PUT	`/v2.1/{tenant_id}/servers/{server_id}`	Updates the editable attributes of a server.

Preconditions

- The server must exist.
- You can edit the `accessIPv4`, `accessIPv6`, `diskConfig` and `name` attributes.

Normal response codes: 200

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

1.2.5.1. Request

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

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

Example 1.7. Update server name: JSON request

```
{
    "server": {
        "name": "new-server-test",
        "imageRef": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "flavorRef": "http://openstack.example.com/flavors/1",
        "metadata": {
            "My Server Name": "Apache1"
        }
    }
}
```

Example 1.8. Update server IP addresses: JSON request

```
{
    "server": {
        "accessIPv4": "192.0.2.0",
        "accessIPv6": "2002:0:0:0:0:0:c000:20e"
    }
}
```

Example 1.9. Update server OS-DCF:diskConfig parameter: JSON request

```
{
    "server": {
        "OS-DCF:diskConfig": "AUTO"
    }
}
```

1.2.5.2. Response

Example 1.10. Update server name: JSON response

```
{
    "server": {
        "id": "52415800-8b69-11e0-9b19-734f565bc83b",
        "tenant_id": "1234",
        "user_id": "5678",
        "name": "new-server-test",
        "created": "2010-11-11T12:00:00Z",
        "updated": "2010-11-12T12:44:44Z",
        "hostId": "e4d909c290d0fb1ca068ffaddf22cbd0",
        "accessIPv4": "192.0.2.0",
        "accessIPv6": "2002:0:0:0:0:0:c000:20e",
        "progress": 0,
        "status": "ACTIVE",
        "image": {
            "id": "52415800-8b69-11e0-9b19-734f6f006e54",
            "name": "CentOS 5.2",
            "links": [
                {
                    "rel": "self",
                    "href": "http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"
                },
                {
                    "rel": "bookmark",
                    "href": "http://servers.api.openstack.org/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"
                }
            ]
        },
        "flavor": {
            "id": "52415800-8b69-11e0-9b19-734f1195ff37",
            "name": "256 MB Server",
            "links": [
                {
                    "rel": "self",
                    "href": "http://servers.api.openstack.org/v2/1234/flavors/
52415800-8b69-11e0-9b19-734f1195ff37"
                },
                {
                    "rel": "bookmark",
                    "href": "http://servers.api.openstack.org/1234/flavors/
52415800-8b69-11e0-9b19-734f1195ff37"
                }
            ]
        },
        "metadata": {
            "My Server Name": "Apache1"
        },
        "addresses": {
            "public": [
                {
                    "version": 4,
                    "addr": "192.0.2.0"
                },
                {
                    "version": 6,
                    "addr": "2002:0:0:0:0:0:c000:20e"
                }
            ],
            "private": [
                {
                    "version": 4,
                    "addr": "198.51.100.0"
                },
                {
                    "version": 6,
                    "addr": "2002:0:0:0:0:0:c633:640e"
                }
            ]
        },
        "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 1.11. Update server IP addresses: JSON response

```
{
    "server": {
        "id": "52415800-8b69-11e0-9b19-734f565bc83b",
        "tenant_id": "1234",
        "user_id": "5678",
        "name": "new-server-test",
        "created": "2010-11-11T12:00:00Z",
        "updated": "2010-11-12T12:55:55Z",
        "hostId": "e4d909c290d0fb1ca068ffaddf22cbd0",
        "accessIPv4": "192.0.2.0",
        "accessIPv6": "2002:0:0:0:0:0:c000:20e",
        "progress": 0,
        "status": "ACTIVE",
        "image": {
            "id": "52415800-8b69-11e0-9b19-734f6f006e54",
            "name": "CentOS 5.2",
            "links": [
                {
                    "rel": "self",
                    "href": "http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"
                },
                {
                    "rel": "bookmark",
                    "href": "http://servers.api.openstack.org/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"
```

```
                }
            ]
        },
        "flavor": {
            "id": "52415800-8b69-11e0-9b19-734f1195ff37",
            "name": "256 MB Server",
            "links": [
                {
                    "rel": "self",
                    "href": "http://servers.api.openstack.org/v2/1234/flavors/
52415800-8b69-11e0-9b19-734f1195ff37"
                },
                {
                    "rel": "bookmark",
                    "href": "http://servers.api.openstack.org/1234/flavors/
52415800-8b69-11e0-9b19-734f1195ff37"
                }
            ]
        },
        "metadata": {
            "My Server Name": "Apache1"
        },
        "addresses": {
            "public": [
                {
                    "version": 4,
                    "addr": "192.0.2.0"
                },
                {
                    "version": 6,
                    "addr": "2002:0:0:0:0:0:c000:20e"
                }
            ],
            "private": [
                {
                    "version": 4,
                    "addr": "198.51.100.0"
                },
                {
                    "version": 6,
                    "addr": "2002:0:0:0:0:0:c633:640e"
                }
            ]
        },
        "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"
            }
        ]
    }
}
```

1.2.6. Delete server

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/servers/{server_id}`	Deletes a server.

Preconditions

- The server must exist.
- Anyone can delete a server when the status of the server is not locked and when the policy allows.
- If the server is locked, you must have administrator privileges to delete the server.

Asynchronous postconditions

- With correct permissions, you can see the server status as DELETED through API calls.
- The port attached to the server is deleted.
- The server does not appear in the list servers response.
- The server managed by OpenStack Compute is deleted on the compute node.

Troubleshooting

- If server status remains in `deleting` status or another error status, the request failed. Ensure that you meet the preconditions. Then, investigate the compute back end.
- The request returns the HTTP 409 response code when the server is locked even if you have correct permissions. Ensure that you meet the preconditions then investigate the server status.
- The server managed by OpenStack Compute is not deleted from the compute node.

Normal response codes: 204

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

1.2.6.1. Request

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

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

This operation does not accept a request body.

1.3. Servers multiple create (servers)

Creates one or more servers.

Optionally, you can set `"return_reservation_id": "True"` in the request body to request that a reservation ID be returned instead of the newly created instance information. With this parameter, the response shows only the reservation ID.

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers`	Creates one or more servers.
POST	`/v2.1/{tenant_id}/servers`	Creates one or more servers with a reservation ID.

1.3.1. Create multiple servers

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

Normal response codes: 202

1.3.1.1. Request

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

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

This list shows the body parameters for the request:

- **parameters**:

  - **block_device_mapping_v2**:

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

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

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

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

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

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

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

      To delete the boot volume when the server is destroyed, specify `true`. Otherwise, specify `false`.

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

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

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

      Defines the order in which a hypervisor tries devices when it attempts to boot the guest from storage.

      Give each device a unique boot index starting from `0`. To disable a device from booting, set the boot index to a negative value or use the default boot index value, which is `None`.

The simplest usage is, set the boot index of the boot device to `0` and use the default boot index value, `None`, for any other devices.

Some hypervisors might not support booting from multiple devices; these hypervisors consider only the device with a boot index of `0`.

Some hypervisors support booting from multiple devices but only if the devices are of different types. For example, a disk and CD-ROM.

Example 1.12. Create multiple servers without reservation ID

```
{
    "server": {
        "name": "new-server-test",
        "imageRef": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "flavorRef": "http://openstack.example.com/openstack/flavors/1",
        "metadata": {
            "My Server Name": "Apache1"
        },
        "min_count": "2",
        "max_count": "3"
    }
}
```

1.3.1.2. Response

Example 1.13. Create multiple servers without reservation ID

```
{
    "server": {
        "OS-DCF:diskConfig": "AUTO",
        "adminPass": "zPnp2GseTqG4",
        "id": "8195065c-fea4-4d57-b93f-5c5c63fe90e8",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/servers/
8195065c-fea4-4d57-b93f-5c5c63fe90e8",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
8195065c-fea4-4d57-b93f-5c5c63fe90e8",
                "rel": "bookmark"
            }
        ],
        "security_groups": [
            {
                "name": "default"
            }
        ]
    }
}
```

1.3.2. Create multiple servers with reservation ID

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers`	Creates one or more servers with a reservation ID.

Set `"return_reservation_id": "True"` in the request body to request that a reservation ID be returned instead of the newly created instance information. With this parameter, the response shows only the reservation ID.

Normal response codes: 202

1.3.2.1. Request

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

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

This list shows the body parameters for the request:

- **parameters**:
  - **block_device_mapping_v2**:
    - **device_name**: Csapi:string. Required.

      A path to the device for the volume that you want to use to boot the server.
    - **source_type**: Csapi:string. Required.

      The source type of the volume. A valid value is `blank`, `snapshot`, `volume`, or `image`.
    - **destination_type**: Csapi:string. Optional.

      Defines where the volume comes from. A valid value is `local` or `volume`.
    - **delete_on_termination**: Csapi:string. Required.

      To delete the boot volume when the server is destroyed, specify `true`. Otherwise, specify `false`.
    - **guest_format**: Csapi:string. Required.

      Specifies the guest server disk file system format, such as `ephemeral` or `swap`.
    - **boot_index**: Csapi:string. Required.

Defines the order in which a hypervisor tries devices when it attempts to boot the guest from storage.

Give each device a unique boot index starting from `0`. To disable a device from booting, set the boot index to a negative value or use the default boot index value, which is `None`.

The simplest usage is, set the boot index of the boot device to `0` and use the default boot index value, `None`, for any other devices.

Some hypervisors might not support booting from multiple devices; these hypervisors consider only the device with a boot index of `0`.

Some hypervisors support booting from multiple devices but only if the devices are of different types. For example, a disk and CD-ROM.

Example 1.14. Create multiple servers with reservation ID

```
{
    "server": {
        "name": "new-server-test",
        "imageRef": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "flavorRef": "http://openstack.example.com/openstack/flavors/1",
        "metadata": {
            "My Server Name": "Apache1"
        },
        "return_reservation_id": "True",
        "min_count": "2",
        "max_count": "3"
    }
}
```

1.3.2.2. Response

Example 1.15. Create multiple servers with reservation ID

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

1.4. Servers actions (servers, action)

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

You can associate a fixed or floating IP address with a server instance, or disassociate a fixed or floating IP address from a server instance. You can attach a volume to a server.

You can create an image from a server, evacuate a server from a failed host to a new host, and force-delete a server before deferred cleanup. You can lock, pause, reboot, rebuild, rescue, resize, resume, confirm the resize of, revert a pending resize for, shelve, shelf-offload, unshelve, start, stop, unlock, unpause, and unrescue a server. You can also change the password of the server.

You can get an RDP, serial, SPICE, or VNC console for a server.

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Adds a fixed IP address to a server instance, which associates that address with the server. The fixed IP address is retrieved from the network that you specify in the request.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Adds a floating IP address to a server, which associates that address with the server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Attaches a volume to a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Confirms a pending resize action for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Creates an image from a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Evacuates a server from a failed host to a new one.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Force-deletes a server before deferred cleanup.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Locks a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Pauses a server. Changes its status to PAUSED.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Shows console output for a server instance.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Gets an RDP console for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Gets a serial console for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Gets a SPICE console for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Gets a VNC console for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Reboots a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Rebuilds a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Removes, or disassociates, a fixed IP address from a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Removes, or disassociates, a floating IP address from a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Puts a server in rescue mode and changes its status to RESCUE.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resizes a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Restores a previously soft-deleted server instance. You cannot use this method to restore deleted instances.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resumes a suspended server and changes its status to ACTIVE.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Cancels and reverts a pending resize action for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Shelves a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Shelf-offloads, or removes, a shelved server.

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Starts a stopped server and changes its status to `ACTIVE`.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Stops a running server and changes its status to `SHUTOFF`.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Unlocks a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Unpauses a paused server and changes its status to `ACTIVE`.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Unrescues a server. Changes status to `ACTIVE`.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Unshelves, or restores, a shelved server.

1.4.1. Add (associate) fixed IP (addFixedIp action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Adds a fixed IP address to a server instance, which associates that address with the server. The fixed IP address is retrieved from the network that you specify in the request.

Specify the `addFixedIp` action and the network ID in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.4.1.1. Request

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

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

Example 1.16. Add (associate) fixed IP (addFixedIp action): JSON request

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

1.4.2. Add (associate) floating IP (addFloatingIp action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Adds a floating IP address to a server, which associates that address with the server.

A pool of floating IP addresses, configured by the cloud administrator, is available in OpenStack Compute. The project quota defines the maximum number of floating IP addresses that you can allocate to the project. After you create (allocate) a floating IP address for a project, you can associate that address with the server. Specify the `addFloatingIp` action in the request body.

If an instance is connected to multiple networks, you can associate a floating IP address with a specific fixed IP address by using the optional `fixed_address` parameter.

Preconditions

- The server must exist.
- You can only add a floating IP address to the server when its status is `available`.

Normal response codes: 202

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

1.4.2.1. Request

This table shows the URI parameters for the add (associate) floating ip (addfloatingip action) request:

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

Example 1.17. Add (associate) floating IP (addFloatingIp action): JSON request

```
{
    "addFloatingIp": {
        "address": "172.24.4.4"
    }
}
```

1.4.3. Attach volume (attach action)

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

Specify the `attach` action in the request body.

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

Preconditions

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

Troubleshooting

- If the request fails due to an OpenStack Compute service error, ensure you meet the preconditions and run the request again. If the request fails again, investigate OpenStack Compute service or ask your cloud provider.

Normal response codes: 202

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

1.4.3.1. Request

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

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

Example 1.18. Attach volume (attach action): JSON request

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

1.4.4. Confirm resized server (confirmResize action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/ {server_id}/action	Confirms a pending resize action for a server.

Specify the confirmResize action in the request body.

After you make this request, you typically must keep polling the server status to determine whether the request succeeded. A successfully confirming resize operation shows a status of ACTIVE or SHUTOFF and a migration_status of confirmed. You can also see the resized server in the compute node that OpenStack Compute manages.

Preconditions

- You can only confirm the resized server where the status is VERIFY_RESIZED, the vm_status is RESIZED, and the migration_status is finished or confirming.
- If the server is locked, you must have administrator privileges to confirm the server.

Troubleshooting

- If the server status remains RESIZED, the request failed. Ensure you meet the preconditions and run the request again. If the request fails again, investigate the compute back end or ask your cloud provider.

Normal response codes: 204

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

1.4.4.1. Request

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

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

Example 1.19. Confirm resized server (confirmResize action): JSON request

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

1.4.5. Create image (createImage action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Creates an image from a server.

Specify the `createImage` action in the request body.

After you make this request, you typically must keep polling the status of the created image to determine whether the request succeeded.

If the operation succeeds, the created image has a status of `active` and the server status returns to the original status. You can also see the new image in the image back end that OpenStack Image service manages.

Preconditions

- The server must exist.
- You can only create a new image from the server when its status is `ACTIVE`, `SHUTOFF`, `PAUSED`, or `SUSPENDED`.
- The connection to the Image service is valid.

Troubleshooting

- If the image status remains uploading or shows another error status, the request failed. Ensure you meet the preconditions and run the request again. If the request fails again, investigate the image back end.
- If the server status does not go back to an original server's status, the request failed. Ensure you meet the preconditions, or check if there is another operation that causes race conditions for the server, then run the request again. If the request fails again, investigate the compute back end or ask your cloud provider.
- If the request fails due to an error on OpenStack Compute service, the image is purged from the image store that OpenStack Image service manages. Ensure you meet the preconditions and run the request again. If the request fails again, investigate OpenStack Compute service or ask your cloud provider.

Normal response codes: 202

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

1.4.5.1. Request

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

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

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

Example 1.20. Create image (createImage action): JSON request

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

1.4.6. Evacuate server (evacuate action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Evacuates a server from a failed host to a new one.

Specify the `evacuate` action in the request body.

Normal response codes: 202

1.4.6.1. Request

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

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

Example 1.21. Evacuate server (evacuate action): JSON request

```
{
    "evacuate": {
        "host": "b419863b7d814906a68fb31703c0dbd6",
        "adminPass": "MySecretPass"
    }
}
```

1.4.6.2. Response

Example 1.22. Evacuate server (evacuate action): JSON response

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

1.4.7. Force-delete server (forceDelete action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Force-deletes a server before deferred cleanup.

Specify the `forceDelete` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.4.7.1. Request

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

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

Example 1.23. Force-delete server (forceDelete action): JSON request

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

1.4.8. Lock server (lock action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Locks a server.

Specify the `lock` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.4.8.1. Request

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

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

Example 1.24. Lock server (lock action): JSON request

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

1.4.9. Pause server (pause action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Pauses a server. Changes its status to PAUSED.

Specify the `pause` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.4.9.1. Request

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

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

Example 1.25. Pause server (pause action): JSON request

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

1.4.10. Show console output (os-getConsoleOutput action)

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

Specify the `os-getConsoleOutput` action in the request body.

Normal response codes: 200

1.4.10.1. Request

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

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

Example 1.26. Get console output: JSON request

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

1.4.10.2. Response

Example 1.27. Get console output: JSON response

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

1.4.11. Get RDP console (os-getRDPConsole action)

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

Specify the `os-getRDPConsole` action in the request body.

Normal response codes: 200

1.4.11.1. Request

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

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

Example 1.28. Get RDP console (os-getRDPConsole action): JSON request

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

1.4.11.2. Response

Example 1.29. Get RDP console (os-getRDPConsole action): JSON response

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

1.4.12. Get serial console (os-getSerialConsole action)

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

Specify the `os-getSerialConsole` action in the request body.

Normal response codes: 200

1.4.12.1. Request

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

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

Example 1.30. Get serial console (os-getSerialConsole action): JSON request

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

1.4.12.2. Response

Example 1.31. Get serial console (os-getSerialConsole action): JSON response

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

1.4.13. Get SPICE console (os-getSPICEConsole action)

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

Specify the `os-getSPICEConsole` action in the request body.

Normal response codes: 200

1.4.13.1. Request

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

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

Example 1.32. Get SPICE console (os-getSPICEConsole action): JSON request

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

1.4.13.2. Response

Example 1.33. Get SPICE console (os-getSPICEConsole action): JSON response

```
{
    "console": {
        "type": "spice-html5",
        "url": "http://127.0.0.1:6082/spice_auto.html?token=
a30e5d08-6a20-4043-958f-0852440c6af4"
    }
}
```

1.4.14. Get VNC console (os-getVNCConsole action)

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

Specify the `os-getVNCConsole` action in the request body.

Normal response codes: 200

1.4.14.1. Request

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

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

Example 1.34. Get VNC console (os-getVNCConsole action): JSON request

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

1.4.14.2. Response

Example 1.35. Get VNC console (os-getVNCConsole action): JSON response

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

1.4.15. Reboot server (reboot action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Reboots a server.

Specify the `reboot` action in the request body.

Normal response codes: 202

Error response codes: computeFault (400, 500, …), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), badMediaType (415), NetworkNotFound (400), HTTPUnprocessableEntity (422), buildInProgress (409)

1.4.15.1. Request

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

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

Example 1.36. Reboot server (reboot action): JSON request

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

1.4.16. Rebuild server (rebuild action)

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

Specify the `rebuild` action in the request body.

To rebuild the server with preservation of the ephemeral partition, set the `preserve_ephemeral` parameter to `true`.

Normal response codes: 202

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

1.4.16.1. Request

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

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

Example 1.37. Rebuild server (rebuild action): JSON request

```
{
    "rebuild": {
        "imageRef": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "name": "foobar",
        "adminPass": "seekr3t",
        "metadata": {
            "meta_var": "meta_val"
        },
        "personality": [
            {
                "path": "/etc/banner.txt",
                "contents": "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
 dCBtb3ZlcyBpbiBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
 IGF0IHN1Y2ggYSBzcGVlZC4uLkl0IGZlZWxzIGFuIGltcHVs
 c2lvbi4uLnRoaXMgaXMgdGhlIHBsYWNlIHRvIGdvIG5vdy4g
 QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
 ZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3VkcywgYW5kIHlv
 dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlmdCB5b3Vy
 c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
 b25zLiINCg0KLVJpY2hhcmQgQmFjaA=="
            }
        ],
        "preserve_ephemeral": true
    }
}
```

1.4.16.2. Response

Example 1.38. Rebuild server (rebuild action): JSON response

```
{
    "server": {
        "accessIPv4": "192.0.2.0",
        "accessIPv6": "2002:0:0:0:0:0:c000:20e",
        "addresses": {
            "private": [
                {
                    "addr": "192.168.0.3",
                    "version": 4
                }
            ]
        },
        "adminPass": "seekr3t",
        "created": "2013-11-14T06:29:00Z",
        "flavor": {
            "id": "1",
            "links": [
                {
                    "href": "http://openstack.example.com/openstack/flavors/
1",
                    "rel": "bookmark"
                }
            ]
        },
        "hostId": "28d8d56f0e3a77e20891f455721cbb68032e017045e20aa5dfc6cb66",
        "id": "a0a80a94-3d81-4a10-822a-daa0cf9e870b",
        "image": {
            "id": "70a599e0-31e7-49b7-b260-868f441e862b",
            "links": [
                {
                    "href": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                    "rel": "bookmark"
                }
            ]
        },
        "links": [
            {
                "href": "http://openstack.example.com/v2.1/servers/
a0a80a94-3d81-4a10-822a-daa0cf9e870b",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
a0a80a94-3d81-4a10-822a-daa0cf9e870b",
                "rel": "bookmark"
            }
        ],
        "metadata": {
            "meta_var": "meta_val"
        },
        "name": "foobar",
        "progress": 0,
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2013-11-14T06:29:02Z",
        "user_id": "fake"
    }
}
```

1.4.17. Remove (disassociate) fixed IP (removeFixedIp action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Removes, or disassociates, a fixed IP address from a server.

Specify the `removeFixedIp` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.4.17.1. Request

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

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

Example 1.39. Remove (disassociate) fixed IP (removeFixedIp action): JSON request

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

1.4.18. Remove (disassociate) floating IP (removeFloatingIp action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Removes, or disassociates, a floating IP address from a server.

The IP address is returned to the pool of IP addresses that is available for all projects. When you remove a floating IP address and that IP address is still associated with a running instance, it is automatically disassociated from that instance.

Specify the `removeFloatingIp` action in the request body.

Normal response codes: 202

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

1.4.18.1. Request

This table shows the URI parameters for the remove (disassociate) floating ip (removefloatingip action) request:

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

Example 1.40. Remove (disassociate) floating IP (removeFloatingIp action): JSON request

```
{
    "removeFloatingIp": {
        "address": "172.24.4.4"
    }
}
```

1.4.19. Rescue server (rescue action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Puts a server in rescue mode and changes its status to RESCUE.

Specify the `rescue` action in the request body.

If you specify the `rescue_image_ref` extended attribute, the image is used to rescue the instance. If you omit an image reference, the base image reference is used by default.

Asynchronous Postconditions

- After you successfully rescue a server and make a GET `/v2.1/{tenant_id}/servers/{server_id}` request, its status changes to RESCUE.

Normal response codes: 202

1.4.19.1. Request

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

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

Example 1.41. Rescue server: JSON request

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

Example 1.42. Rescue server with image: JSON request

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

1.4.19.2. Response

Example 1.43. Extended rescue server: JSON response

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

1.4.20. Resize server (resize action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resizes a server.

Specify the `resize` action in the request body.

A successfully resized server shows a `VERIFY_RESIZE` status, `RESIZED` VM status, and `finished` migration status. If you set the `auto_confirm` option of the Compute service to `True`, the Compute service automatically confirms the resize operation.

Preconditions

- You can only resize a server when its status is `ACTIVE` or `SHUTOFF`.
- If the server is locked, you must have administrator privileges to resize the server.

Normal response codes: 202

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

1.4.20.1. Request

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

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

Example 1.44. Resize server (resize action): JSON request

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

1.4.21. Restore soft-deleted instance (restore action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Restores a previously soft-deleted server instance. You cannot use this method to restore deleted instances.

Specify the `restore` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.4.21.1. Request

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

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

Example 1.45. Restore soft-deleted instance (restore action): JSON request

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

1.4.22. Resume suspended server (resume action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resumes a suspended server and changes its status to ACTIVE.

Specify the `resume` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.4.22.1. Request

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

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

Example 1.46. Resume suspended server (resume action): JSON request

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

1.4.23. Revert resized server (revertResize action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Cancels and reverts a pending resize action for a server.

Specify the `revertResize` action in the request body.

After you make this request, you typically must keep polling the server status to determine whether the request succeeded. A successfully reverting resize operation shows a status of `ACTIVE` or `SHUTOFF` and a migration_status of `reverted`. You can also see the reverted server in the compute node that OpenStack Compute manages.

Preconditions

- You can only confirm the resized server where the status is `VERIFY_RESIZE` and the vm_status is `RESIZED`.
- If the server is locked, you must have administrator privileges to revert the resizing.

Troubleshooting

- If the server status remains `RESIZED`, the request failed. Ensure you meet the preconditions and run the request again. If the request fails again, investigate the compute back end.
- The server is not reverted in the compute node that OpenStack Compute manages.

Normal response codes: 202

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

1.4.23.1. Request

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

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

Example 1.47. Revert resized server (revertResize action): JSON request

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

1.4.24. Shelve server (shelve action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Shelves a server.

Specify the `shelve` action in the request body.

All associated data and resources are kept but anything still in memory is not retained. To restore a shelved instance, use the `unshelve` action. To remove a shelved instance, use the `shelveOffload` action.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Preconditions

- The server status must be `ACTIVE`, `SHUTOFF`, `PAUSED`, or `SUSPENDED`.
- If the server is locked, you must have administrator privileges to shelve the server.

Asynchronous Postconditions

- After you successfully shelve a server, its status changes to `SHELVED` and the image status is `ACTIVE`. The server instance data appears on the compute node that the Compute service manages.
- If you boot the server from volumes or set the `shelved_offload_time` option to 0, the Compute service automatically deletes the instance on compute nodes and changes the server status to `SHELVED_OFFLOADED`.

Troubleshooting

- If the server status does not change to `SHELVED` or `SHELVED_OFFLOADED`, the shelve operation failed. Ensure that you meet the preconditions and run the request again. If the request fails again, investigate whether another operation is running that causes a race condition.

Normal response codes: 202

1.4.24.1. Request

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

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

Example 1.48. Shelve server: JSON request

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

1.4.25. Shelf-offload (remove) server (shelveOffload action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Shelf-offloads, or removes, a shelved server.

Specify the `shelveOffload` action in the request body.

Data and resource associations are deleted. If an instance is no longer needed, you can remove that instance from the hypervisor to minimize resource usage.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Preconditions

- The server status must be `SHELVED`.
- If the server is locked, you must have administrator privileges to shelve-offload the server.

Asynchronous Postconditions

- After you successfully shelve-offload a server, its status changes to `SHELVED_OFFLOADED`. The server instance data appears on the compute node.

Troubleshooting

- If the server status does not change to `SHELVED_OFFLOADED`, the shelve-offload operation failed. Ensure that you meet the preconditions and run the request again. If the request fails again, investigate whether another operation is running that causes a race condition.

Normal response codes: 202

1.4.25.1. Request

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

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

Example 1.49. Shelve server: JSON request

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

1.4.26. Start server (os-start action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Starts a stopped server and changes its status to ACTIVE.

Specify the `os-start` action in the request body.

Preconditions

- The server status must be SHUTOFF.
- If the server is locked, you must have administrator privileges to start the server.

Asynchronous Postconditions

- After you successfully start a server, its status changes to ACTIVE. The server appears on the compute node that the Compute service manages.

Troubleshooting

- If the server status does not change to ACTIVE, the start operation failed. Ensure that you meet the preconditions and run the request again. If the request fails again, investigate whether another operation is running that causes a race condition.

Normal response codes: 202

1.4.26.1. Request

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

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

Example 1.50. Start server: JSON request

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

1.4.27. Stop server (os-stop action)

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

Specify the os-stop action in the request body.

Preconditions

- The server status must be ACTIVE or ERROR.
- If the server is locked, you must have administrator privileges to stop the server.

Asynchronous Postconditions

- After you successfully stop a server, its status changes to SHUTOFF. The server instance data appears only on the compute node that Compute service manages.

Normal response codes: 202

1.4.27.1. Request

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

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

Example 1.51. Stop server: JSON request

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

1.4.28. Unlock server (unlock action)

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

Specify the `unlock` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.4.28.1. Request

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

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

Example 1.52. Unlock server (unlock action): JSON request

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

1.4.29. Unpause server (unpause action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Unpauses a paused server and changes its status to ACTIVE.

Specify the `unpause` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.4.29.1. Request

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

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

Example 1.53. Unpause server (unpause action): JSON request

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

1.4.30. Unrescue server (unrescue action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Unrescues a server. Changes status to `ACTIVE`.

Specify the `unrescue` action in the request body.

Preconditions

- The server must exist.
- You can only unrescue a server when its status is RESCUE.

Asynchronous Postconditions

- After you successfully unrescue a server and make a `GET /v2.1/{tenant_id}/servers/{server_id}` request, its status changes to `ACTIVE`.

Normal response codes: 202

1.4.30.1. Request

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

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

Example 1.54. Unrescue server: JSON request

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

1.4.31. Unshelve (restore) shelved server (unshelve action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Unshelves, or restores, a shelved server.

Specify the `unshelve` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Preconditions

- The server status must be `SHELVED` or `SHELVED_OFFLOADED`.
- If the server is locked, you must have administrator privileges to unshelve the server.

Asynchronous Postconditions

- After you successfully shelve a server, its status changes to `ACTIVE`. The server appears on the compute node.
- The shelved image is deleted from the list of images returned by an API call.

Troubleshooting

- If the server status does not change to `ACTIVE`, the unshelve operation failed. Ensure that you meet the preconditions and run the request again. If the request fails again, investigate whether another operation is running that causes a race condition.

Normal response codes: 202

1.4.31.1. Request

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

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

Example 1.55. Unshelve server: JSON request

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

1.5. Servers admin actions (servers, action)

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

You can change the administrative password for, create a back up of, and inject network information into a server.

You can migrate, live-migrate, reset networking on, and reset the state of a server.

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Creates a back up of a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Injects network information into a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Migrates a server to a host. The scheduler chooses the host.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Live-migrates a server to a new host without rebooting.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resets networking on a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resets the state of a server.

1.5.1. Create server back up (createBackup action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Creates a back up of a server.

Specify the `createBackup` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.5.1.1. Request

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

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

Example 1.56. Create server back up (createBackup action): JSON request

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

1.5.2. Inject network information (injectNetworkInfo action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Injects network information into a server.

Specify the `injectNetworkInfo` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.5.2.1. Request

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

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

Example 1.57. Inject network information (injectNetworkInfo action): JSON request

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

1.5.3. Migrate server (migrate action)

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/action	Migrates a server to a host. The scheduler chooses the host.

Specify the `migrate` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.5.3.1. Request

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

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

Example 1.58. Migrate server (migrate action): JSON request

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

1.5.4. Live-migrate server (os-migrateLive action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Live-migrates a server to a new host without rebooting.

Use the `host` parameter to specify the destination host. If you omit this parameter, the scheduler chooses a host. If a scheduled host is not suitable, the scheduler tries up to `migrate_max_retries` rescheduling attempts.

If both source and destination hosts provide local disks, you can set the `block_migration` parameter to `True`. If either host uses shared storage, the migration fails if you set this parameter to `True`.

Policy defaults enable only users with the administrative role to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.5.4.1. Request

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

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

Example 1.59. Live-migrate server (os-migrateLive action): JSON request

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

1.5.5. Reset networking on a server (resetNetwork action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resets networking on a server.

Specify the `resetNetwork` action in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.5.5.1. Request

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

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

Example 1.60. Reset networking on a server (resetNetwork action): JSON request

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

1.5.6. Reset server state (os-resetState action)

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/action`	Resets the state of a server.

Specify the `os-resetState` action and the `state` in the request body.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.5.6.1. Request

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

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

Example 1.61. Reset server state (os-resetState action): JSON request

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

1.6. Servers diagnostics (servers, diagnostics)

Shows the usage data for a server.

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

1.6.1. Show server diagnostics

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

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.6.1.1. Request

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

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

This operation does not accept a request body.

1.6.1.2. Response

Example 1.62. Server diagnostics: JSON response

```
{
    "cpu0_time": 17300000000,
    "memory": 524288,
    "vda_errors": -1,
    "vda_read": 262144,
    "vda_read_req": 112,
    "vda_write": 5778432,
    "vda_write_req": 488,
    "vnet1_rx": 2070139,
    "vnet1_rx_drop": 0,
    "vnet1_rx_errors": 0,
    "vnet1_rx_packets": 26701,
    "vnet1_tx": 140208,
    "vnet1_tx_drop": 0,
    "vnet1_tx_errors": 0,
    "vnet1_tx_packets": 662
}
```

1.7. Servers IPs (servers, ips)

Lists the IP addresses for an instance and shows details for an IP address.

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/ips`	Lists IP addresses that are assigned to an instance.
GET	`/v2.1/{tenant_id}/servers/{server_id}/ips/{network_label}`	Shows IP addresses details for a network label of a server instance.

1.7.1. List IPs

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/ips`	Lists IP addresses that are assigned to an instance.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.7.1.1. Request

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

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

This operation does not accept a request body.

1.7.1.2. Response

Example 1.63. List IPs: JSON response

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

1.7.2. Show IP details

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/ips/{network_label}`	Shows IP addresses details for a network label of a server instance.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.7.2.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{server_id}`	UUID	The UUID of the server.
`{network_label}`	String	The network label, such as `public` or `private`.

This operation does not accept a request body.

1.7.2.2. Response

Example 1.64. Show IP details: JSON response

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

1.8. Server metadata (servers, metadata)

Lists metadata, creates or replaces one or more metadata items, and updates one or more metadata items for a server.

Shows details for, creates or replaces, and updates a metadata item, by key, for a server.

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/metadata`	Lists all metadata for a server.
PUT	`/v2.1/{tenant_id}/servers/{server_id}/metadata`	Creates or replaces one or more metadata items for a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/metadata`	Updates one or more metadata items for a server.
GET	`/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}`	Shows details for a metadata item, by key, for a server.

Method	URI	Description
PUT	`/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}`	Creates or replaces a metadata item, by key, for a server.
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}`	Deletes a metadata item, by key, from a server.

1.8.1. List all metadata

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/metadata`	Lists all metadata for a server.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.8.1.1. Request

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

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

This operation does not accept a request body.

1.8.1.2. Response

Example 1.65. List all metadata: JSON response

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

1.8.2. Create or replace metadata items

Method	URI	Description
PUT	/v2.1/{tenant_id}/servers/{server_id}/metadata	Creates or replaces one or more metadata items for a server.

Creates any metadata items that do not already exist in the server. Removes and completely replaces any metadata items that already exist in the server with the metadata items in the request.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.8.2.1. Request

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

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

Example 1.66. Create or replace metadata items: JSON request

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

1.8.2.2. Response

Example 1.67. Create or replace metadata items: JSON response

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

1.8.3. Update metadata items

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/metadata`	Updates one or more metadata items for a server.

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

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.8.3.1. Request

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

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

Example 1.68. Update metadata items: JSON request

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

1.8.3.2. Response

Example 1.69. Update metadata items: JSON response

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

1.8.4. Show metadata item details

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}`	Shows details for a metadata item, by key, for a server.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.8.4.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{server_id}`	UUID	The UUID of the server.
`{key}`	String	The metadata item key, as a string. Maximum length is 255 characters.

This operation does not accept a request body.

1.8.4.2. Response

Example 1.70. Show metadata item details: JSON response

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

1.8.5. Create or update metadata item

Method	URI	Description
PUT	`/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}`	Creates or replaces a metadata item, by key, for a server.

Creates a metadata item that does not already exist in the server. Removes and completely replaces a metadata item that already exists in the server with the metadata item in the request.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.8.5.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{server_id}`	UUID	The UUID of the server.
`{key}`	String	The metadata item key, as a string. Maximum length is 255 characters.

Example 1.71. Create or update metadata item: JSON request

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

1.8.5.2. Response

Example 1.72. Create or update metadata item: JSON response

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

1.8.6. Delete metadata item

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/metadata/{key}`	Deletes a metadata item, by key, from a server.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 204

1.8.6.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{server_id}`	UUID	The UUID of the server.
`{key}`	String	The metadata item key, as a string. Maximum length is 255 characters.

This operation does not accept a request body.

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

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

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

1.9.1. List actions for server

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-instance-actions`	Lists actions for a server.

Normal response codes: 200

1.9.1.1. Request

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

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

This operation does not accept a request body.

1.9.1.2. Response

Example 1.73. List actions for server: JSON response

```
{
    "instanceActions": [
        {
            "action": "resize",
            "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
            "message": "",
            "project_id": "842",
            "request_id": "req-25517360-b757-47d3-be45-0e8d2a01b36a",
            "start_time": "2012-12-05T01:00:00.000000",
            "user_id": "789"
        },
        {
            "action": "reboot",
            "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
            "message": "",
            "project_id": "147",
            "request_id": "req-3293a3f1-b44c-4609-b8d2-d81b105636b8",
            "start_time": "2012-12-05T00:00:00.000000",
            "user_id": "789"
        }
    ]
}
```

1.9.2. Show server action details

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

Normal response codes: 200

1.9.2.1. Request

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

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

This operation does not accept a request body.

1.9.2.2. Response

Example 1.74. Show server action details: JSON response

```
{
    "instanceAction": {
        "action": "reboot",
        "events": [
            {
                "event": "schedule",
                "finish_time": "2012-12-05T01:02:00.000000",
                "result": "Success",
                "start_time": "2012-12-05T01:00:02.000000",
                "traceback": ""
            },
            {
                "event": "compute_create",
                "finish_time": "2012-12-05T01:04:00.000000",
                "result": "Success",
                "start_time": "2012-12-05T01:03:00.000000",
                "traceback": ""
            }
        ],
        "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
        "message": "",
        "project_id": "147",
        "request_id": "req-3293a3f1-b44c-4609-b8d2-d81b105636b8",
        "start_time": "2012-12-05T00:00:00.000000",
        "user_id": "789"
    }
}
```

1.10. Port interfaces (servers, os-interface)

Creates a port interface and uses it to attach a port to a server and detaches a port interface from a server. Also, lists all port interfaces and shows details for a port interface.

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-interface`	Creates a port interface and uses it to attach a port to a server instance.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-interface`	Lists port interfaces that are attached to a server.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}`	Shows details for a port interface that is attached to a server.
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}`	Detaches a port interface.

1.10.1. Create interface

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-interface`	Creates a port interface and uses it to attach a port to a server instance.

Normal response codes: 200

Error response codes: computeFault (400, 500, …), computeFault (400, 500, …), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), itemNotFound (404), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), badMediaType (415), NetworkNotFound (400)

1.10.1.1. Request

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

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

Example 1.75. Create interface: JSON request

```
{
    "interfaceAttachment": {
        "port_id": "ce531f90-199f-48c0-816c-13e38010b442"
    }
}
```

1.10.1.2. Response

Example 1.76. Create interface: JSON response

```
{
    "interfaceAttachment": {
        "fixed_ips": [
            {
                "ip_address": "192.168.1.3",
                "subnet_id": "f8a6e8f8-c2ec-497c-9f23-da9616de54ef"
            }
        ],
        "mac_addr": "fa:16:3e:4c:2c:30",
        "net_id": "3cb9bc59-5699-4588-a4b1-b87f96708bc6",
        "port_id": "ce531f90-199f-48c0-816c-13e38010b442",
        "port_state": "ACTIVE"
    }
}
```

1.10.2. List port interfaces

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-interface`	Lists port interfaces that are attached to a server.

Normal response codes: 200

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

1.10.2.1. Request

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

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

This operation does not accept a request body.

1.10.2.2. Response

Example 1.77. List port interfaces: JSON response

```
{
    "interfaceAttachments": [
        {
            "fixed_ips": [
                {
                    "ip_address": "192.168.1.3",
                    "subnet_id": "f8a6e8f8-c2ec-497c-9f23-da9616de54ef"
                }
            ],
            "mac_addr": "fa:16:3e:4c:2c:30",
            "net_id": "3cb9bc59-5699-4588-a4b1-b87f96708bc6",
            "port_id": "ce531f90-199f-48c0-816c-13e38010b442",
            "port_state": "ACTIVE"
        }
    ]
}
```

1.10.3. Show port interface details

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}	Shows details for a port interface that is attached to a server.

Normal response codes: 200

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

1.10.3.1. Request

This table shows the URI parameters for the show port interface details request:

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

This operation does not accept a request body.

1.10.3.2. Response

Example 1.78. Show port interface details: JSON response

```
{
    "interfaceAttachment": {
        "fixed_ips": [
            {
                "ip_address": "192.168.1.3",
                "subnet_id": "f8a6e8f8-c2ec-497c-9f23-da9616de54ef"
            }
        ],
        "mac_addr": "fa:16:3e:4c:2c:30",
        "net_id": "3cb9bc59-5699-4588-a4b1-b87f96708bc6",
        "port_id": "ce531f90-199f-48c0-816c-13e38010b442",
        "port_state": "ACTIVE"
    }
}
```

1.10.4. Detach interface

Method	URI	Description
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-interface/{attachment_id}	Detaches a port interface.

Normal response codes: 202

Error response codes: computeFault (400, 500, …), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), badMediaType (415), NetworkNotFound (400)

1.10.4.1. Request

This table shows the URI parameters for the detach interface request:

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

This operation does not accept a request body.

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

Shows the encrypted administrative password. Also, clears the encrypted administrative password for a server, which removes it from the metadata server.

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-server-password	Shows the administrative password for a server.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-server-password	Clears the encrypted administrative password for a server, which removes it from the metadata server.

1.11.1. Show server password

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

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.11.1.1. Request

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

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

This operation does not accept a request body.

1.11.1.2. Response

Example 1.79. Show server password: JSON response

```
{
    "password": "xlozO3wLCBRWAa2yDjCCVx8vwNPypxnypmRYDa/zErlQ+EzPe1S/
Gz6nfmC52mOlOSCRuUOmG7kqqgejPof6M7bOezS387zjq4LSvvwp28zUknzy4YzfFGhnHAdai3TxUJ26pfQCYrq8UTzm
I1K2LsuipfxSJR7Wdke4zNXJjHHP2RfYsVbZ/k9ANu+Nz4iIH8/7Cacud/
pphH7EjrY6a4RZNrjQskrhKYed0YERpotyjYk1eDtRe72GrSiXteqCM4biaQ5w3ruS+AcX//
PXk3uJ5kC7d67fPXaVz4WaQRYMg=="
}
```

1.11.2. Clear admin password

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/os-server-password`	Clears the encrypted administrative password for a server, which removes it from the metadata server.

This action does not actually change the instance server password.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 204

1.11.2.1. Request

This table shows the URI parameters for the clear admin password request:

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

This operation does not accept a request body.

1.12. Servers virtual interfaces (servers, os-virtual-interfaces)

Lists virtual interfaces for a server instance.

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-virtual-interfaces`	Lists the virtual interfaces for an instance.

1.12.1. List virtual interfaces

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Lists the virtual interfaces for an instance.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Change these permissions through the `policy.json` file.

The API v2 returns the network ID in the `OS-EXT-VIF-NET:net_id` response attribute.

The API v2.1 base version does not return the network ID.

Normal response codes: 200

1.12.1.1. Request

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

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

This operation does not accept a request body.

1.12.1.2. Response

Example 1.80. List virtual interfaces: JSON response

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

1.13. Flavors with extended attributes (flavors)

Shows information about flavors.

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

1.13.1. List flavors

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

Normal response codes: 200

1.13.1.1. Request

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

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

1.13.1.2. Response

Example 1.81. List flavors: JSON response

```
{
    "flavors": [
        {
            "id": "1",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/openstack/
flavors/1",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/flavors/
1",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.tiny"
        },
        {
            "id": "2",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/openstack/
flavors/2",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/flavors/
2",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.small"
        },
        {
            "id": "3",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/openstack/
flavors/3",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/flavors/
3",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.medium"
        },
        {
            "id": "4",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/openstack/
flavors/4",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/flavors/
4",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.large"
        },
        {
            "id": "5",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/openstack/
flavors/5",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/flavors/
5",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.xlarge"
        }
    ]
}
```

1.13.2. Show flavor details

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

Normal response codes: 200

1.13.2.1. Request

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

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

1.13.2.2. Response

Example 1.82. Show flavor details: JSON response

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

1.13.3. List flavors with details

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

Normal response codes: 200

1.13.3.1. Request

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

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

1.13.3.2. Response

Example 1.83. List flavors with details: JSON response

```
{
    "flavors": [
        {
            "OS-FLV-DISABLED:disabled": false,
            "disk": 1,
            "OS-FLV-EXT-DATA:ephemeral": 0,
            "os-flavor-access:is_public": true,
            "id": "1",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/openstack/
flavors/1",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/flavors/
1",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.tiny",
            "ram": 512,
            "swap": "",
            "vcpus": 1
        },
        {
            "OS-FLV-DISABLED:disabled": false,
            "disk": 20,
            "OS-FLV-EXT-DATA:ephemeral": 0,
            "os-flavor-access:is_public": true,
            "id": "2",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/openstack/
flavors/2",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/flavors/
2",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.small",
            "ram": 2048,
            "swap": "",
            "vcpus": 1
        },
        {
            "OS-FLV-DISABLED:disabled": false,
            "disk": 40,
            "OS-FLV-EXT-DATA:ephemeral": 0,
            "os-flavor-access:is_public": true,
            "id": "3",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/openstack/
flavors/3",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/flavors/
3",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.medium",
            "ram": 4096,
            "swap": "",
            "vcpus": 2
        },
        {
            "OS-FLV-DISABLED:disabled": false,
            "disk": 80,
            "OS-FLV-EXT-DATA:ephemeral": 0,
            "os-flavor-access:is_public": true,
            "id": "4",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/openstack/
flavors/4",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/flavors/
4",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.large",
            "ram": 8192,
            "swap": "",
            "vcpus": 4
        },
        {
            "OS-FLV-DISABLED:disabled": false,
            "disk": 160,
            "OS-FLV-EXT-DATA:ephemeral": 0,
            "os-flavor-access:is_public": true,
            "id": "5",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/openstack/
flavors/5",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/openstack/flavors/
5",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.xlarge",
            "ram": 16384,
            "swap": "",
            "vcpus": 8
        }
    ]
}
```

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

Provides flavor access support.

Method	URI	Description
GET	`/v2.1/{tenant_id}/flavors/os-flavor-access`	Lists flavor access information.
POST	`/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action`	Adds flavor access to a tenant and flavor.
POST	`/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action`	Removes flavor access from a tenant and flavor.

1.14.1. List flavor access information for given flavor

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

Normal response codes: 200

1.14.1.1. Request

This table shows the URI parameters for the list flavor access information for given flavor request:

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

This operation does not accept a request body.

1.14.1.2. Response

Example 1.84. List flavor access information for given flavor: JSON response

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

1.14.2. Add flavor access to tenant

Method	URI	Description
POST	/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action	Adds flavor access to a tenant and flavor.

Specify the `add_tenant_access` action and the `tenant_id` in the request body.

Normal response codes: 200

1.14.2.1. Request

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

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

Example 1.85. Add flavor access to tenant: JSON request

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

1.14.2.2. Response

Example 1.86. Add flavor access to tenant: JSON response

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

1.14.3. Remove flavor access from tenant

Method	URI	Description
POST	`/v2.1/{tenant_id}/flavors/os-flavor-access/{flavor_id}/action`	Removes flavor access from a tenant and flavor.

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

Normal response codes: 200

1.14.3.1. Request

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

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

Example 1.87. Remove flavor access from tenant: JSON request

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

1.14.3.2. Response

Example 1.88. Remove flavor access from tenant: JSON response

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

1.15. Flavors extra-specs (flavors, os-flavor-extra-specs)

Lists, creates, deletes, and updates the extra-specs or keys for a flavor.

Method	URI	Description
POST	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs`	Creates extra specs for a flavor, by ID.
GET	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs`	Lists all extra specs for a flavor, by ID.
GET	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}`	Shows an extra spec, by key, for a flavor, by ID.
PUT	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}`	Updates an extra spec, by key, for a flavor, by ID.

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}`	Deletes an extra spec, by key, for a flavor, by ID.

1.15.1. Create extra specs for a flavor

Method	URI	Description
POST	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs`	Creates extra specs for a flavor, by ID.

Normal response codes: 201

1.15.1.1. Request

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

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

Example 1.89. Create extra specs for a flavor: JSON request

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

1.15.1.2. Response

Example 1.90. Create extra specs for a flavor: JSON response

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

1.15.2. List extra specs for a flavor

Method	URI	Description
GET	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs`	Lists all extra specs for a flavor, by ID.

Normal response codes: 200

1.15.2.1. Request

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

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

Example 1.91. List extra specs for a flavor: JSON request

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

1.15.2.2. Response

Example 1.92. List extra specs for a flavor: JSON response

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

1.15.3. Show an extra spec for a flavor

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

Normal response codes: 200

1.15.3.1. Request

This table shows the URI parameters for the show an extra spec for a flavor request:

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

This operation does not accept a request body.

1.15.3.2. Response

Example 1.93. Show an extra spec for a flavor: JSON response

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

1.15.4. Update an extra spec for a flavor

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

Normal response codes: 200

1.15.4.1. Request

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

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

Example 1.94. Update an extra spec for a flavor: JSON request

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

1.15.4.2. Response

Example 1.95. Update an extra spec for a flavor: JSON response

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

1.15.5. Delete an extra spec for a flavor

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{flavor_extra_spec_key}`	Deletes an extra spec, by key, for a flavor, by ID.

Normal response codes: 204

1.15.5.1. Request

This table shows the URI parameters for the delete an extra spec for a flavor request:

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

This operation does not accept a request body.

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

Creates and deletes flavors.

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

1.16.1. Create flavor

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

Normal response codes: 200

1.16.1.1. Request

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

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

Example 1.96. Create flavor: JSON request

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

1.16.1.2. Response

Example 1.97. Create flavor: JSON response

```
{
    "flavor": {
        "OS-FLV-DISABLED:disabled": false,
        "disk": 10,
        "OS-FLV-EXT-DATA:ephemeral": 0,
        "os-flavor-access:is_public": true,
        "id": "10",
        "links": [
            {
                "href": "http://openstack.example.com/v2.1/flavors/10",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/flavors/10",
                "rel": "bookmark"
            }
        ],
        "name": "test_flavor",
        "ram": 1024,
        "swap": "",
        "vcpus": 2
    }
}
```

1.16.2. Delete flavor

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

Normal response codes: 202

1.16.2.1. Request

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

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

This operation does not accept a request body.

1.17. Keypairs (keypairs)

Generates, imports, and deletes SSH keys.

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

1.17.1. List keypairs

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

Normal response codes: 200

1.17.1.1. Request

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

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

This operation does not accept a request body.

1.17.1.2. Response

Example 1.98. List keypairs: JSON response

```
{
    "keypair": {
        "fingerprint": "44:fe:29:6e:23:14:b9:53:5b:65:82:58:1c:fe:5a:c3",
        "name": "keypair-6638abdb-c4e8-407c-ba88-c8dd7cc3c4f1",
        "public_key": "ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQC1HTrHCbb9NawNLSV8N6tSa8i637+EC2dA+lsdHHfQlT54t
+N0nHhJPlKWDLhc579j87vp6RDFriFJ/smsTnDnf64O12z0kBaJpJPH2zXrBkZFK6q2rmxydURzX/
z0yLSCP77SFJ0fdXWH2hMsAusflGyryHGX20n
+mZK6mDrxVzGxEz228dwQ5G7Az5OoZDWygH2pqPvKjkifRw0jwUKf3BbkP0QvANACOk26cv16mNFpFJfI1N3OC5lUsZQ
qup58J5kflNm7I61sy1mJon6SGqNUSfoQagqtBH6vd/tU1jnlwZ03uUroAL Generated-by-Nova\
n",
        "user_id": "fake",
        "deleted": false,
        "created_at": "2014-05-07T12:06:13.681238",
        "updated_at": null,
        "deleted_at": null,
        "id": 1
    }
}
```

1.17.2. Create or import keypair

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

Normal response codes: 200

1.17.2.1. Request

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

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

Example 1.99. Create or import keypair: JSON request

```
{
    "keypair": {
        "name": "keypair-d20a3d59-9433-4b79-8726-20b431d89c78",
        "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDx8nkQv/
zgGgB4rMYmIf+6A4l6Rr+o/6lHBQdW5aYd44bd8JttDCE/F/pNRr0lRE
+PiqSPO8nDPHw0010JeMH9gYgnnFlyY3/OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/
0d3ac30bNH6Sw9JD9UZHYcpSxsIbECHw== Generated-by-Nova"
    }
}
```

1.17.2.2. Response

Example 1.100. Create or import keypair: JSON response

```
{
    "keypair": {
        "fingerprint": "1e:2c:9b:56:79:4b:45:77:f9:ca:7a:98:2c:b0:d5:3c",
        "name": "keypair-803a1926-af78-4b05-902a-1d6f7a8d9d3e",
        "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDx8nkQv/
zgGgB4rMYmIf+6A4l6Rr+o/6lHBQdW5aYd44bd8JttDCE/F/pNRr0lRE
+PiqSPO8nDPHw0010JeMH9gYgnnFlyY3/OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/
0d3ac30bNH6Sw9JD9UZHYcpSxsIbECHw== Generated-by-Nova",
        "user_id": "fake"
    }
}
```

1.17.3. Delete keypair

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-keypairs/{keypair_name}`	Deletes a keypair.

Normal response codes: 204

1.17.3.1. Request

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

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

This operation does not accept a request body.

1.17.4. Show keypair details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-keypairs/{keypair_name}`	Shows details for a keypair that is associated with the account.

Normal response codes: 200

1.17.4.1. Request

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

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

This operation does not accept a request body.

1.17.4.2. Response

Example 1.101. Show keypair details: JSON response

```
{
    "keypairs": [
        {
            "keypair": {
                "fingerprint":
 "7e:eb:ab:24:ba:d1:e1:88:ae:9a:fb:66:53:df:d3:bd",
                "name": "keypair-50ca852e-273f-4cdc-8949-45feba200837",
                "public_key": "ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCkF3MX59OrlBs3dH5CU7lNmvpbrgZxSpyGjlnE8Flkirnc/
Up22lpjznoxqeoTAwTW034k7Dz6aYIrZGmQwe2TkE084yqvlj45Dkyoj95fW/
sZacm0cZNuL69EObEGHdprfGJQajrpz22NQoCD8TFB8Wv+8om9NH9Le6s
+WPe98WC77KLw8qgfQsbIey+JawPWl4O67ZdL5xrypuRjfIPWjgy/
VH85IXg/Z/GONZ2nxHgSShMkwqSFECAC5L3PHB+0+/12M/
iikdatFSVGjpuHvkLOs3oe7m6HlOfluSJ85BzLWBbvva93qkGmLg4ZAc8rPh2O+YIsBUHNLLMM/oQp
 Generated-by-Nova\n"
            }
        }
    ]
}
```

1.18. Limits (limits)

Shows rate and absolute limits for the tenant.

Method	URI	Description
GET	`/v2.1/{tenant_id}/limits`	Shows rate and absolute limits for the tenant.

1.18.1. Show rate and absolute limits

Method	URI	Description
GET	`/v2.1/{tenant_id}/limits`	Shows rate and absolute limits for the tenant.

Normal response codes: 200

1.18.1.1. Request

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

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

This operation does not accept a request body.

1.18.1.2. Response

Example 1.102. Show rate and absolute limits: JSON response

```
{
    "limits": {
        "rate": [],
        "absolute": {
            "maxServerMeta": 128,
            "maxPersonality": 5,
            "totalServerGroupsUsed": 0,
            "maxImageMeta": 128,
            "maxPersonalitySize": 10240,
            "maxTotalKeypairs": 100,
            "maxSecurityGroupRules": 20,
            "maxServerGroups": 10,
            "totalCoresUsed": 1,
            "totalRAMUsed": 2048,
            "totalInstancesUsed": 1,
            "maxSecurityGroups": 10,
            "totalFloatingIpsUsed": 0,
            "maxTotalCores": 20,
            "maxServerGroupMembers": 10,
            "maxTotalFloatingIps": 10,
            "totalSecurityGroupsUsed": 1,
            "maxTotalInstances": 10,
            "maxTotalRAMSize": 51200
        }
    }
}
```

1.19. Extensions (extensions)

Lists available extensions and shows information for an extension, by alias. The OpenStack Compute API v2.0 is extensible. For information about extensions, see Extensions.

Method	URI	Description
GET	`/v2.1/{tenant_id}/extensions`	Lists available extensions.

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

1.19.1. List extensions

Method	URI	Description
GET	`/v2.1/{tenant_id}/extensions`	Lists available extensions.

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

The response shows the extension name and its alias. To show details for an extension, you specify the alias.

Normal response codes: 200203

1.19.1.1. Request

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

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

This operation does not accept a request body.

1.19.1.2. Response

Example 1.103. List extensions: JSON response

```
{
    "extensions": [
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Multinic",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "NMN",
            "description": "Multiple network support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "DiskConfig",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "OS-DCF",
            "description": "Disk Management Extension."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedAvailabilityZone",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "OS-EXT-AZ",
            "description": "Extended Availability Zone support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ImageSize",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "OS-EXT-IMG-SIZE",
            "description": "Adds image size to image listings."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedIps",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "OS-EXT-IPS",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedIpsMac",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "OS-EXT-IPS-MAC",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedServerAttributes",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "OS-EXT-SRV-ATTR",
            "description": "Extended Server Attributes support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedStatus",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "OS-EXT-STS",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FlavorDisabled",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "OS-FLV-DISABLED",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FlavorExtraData",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "OS-FLV-EXT-DATA",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "SchedulerHints",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "OS-SCH-HNT",
            "description": "Pass arbitrary key/value pairs to the scheduler."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ServerUsage",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "OS-SRV-USG",
            "description": "Adds launched_at and terminated_at on Servers."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "AccessIPs",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-access-ips",
            "description": "Access IPs support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "AdminActions",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-admin-actions",
            "description": "Enable admin-only server actions\n\n    Actions
include: resetNetwork, injectNetworkInfo, os-resetState\n    "
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "AdminPassword",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-admin-password",
            "description": "Admin password management support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Agents",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-agents",
            "description": "Agents support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Aggregates",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-aggregates",
            "description": "Admin-only aggregate administration."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "AssistedVolumeSnapshots",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-assisted-volume-snapshots",
            "description": "Assisted volume snapshots."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "AttachInterfaces",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-attach-interfaces",
            "description": "Attach interface support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "AvailabilityZone",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-availability-zone",
            "description": "1. Add availability_zone to the Create Server API.
\n       2. Add availability zones describing.\n    "
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "BareMetalExtStatus",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-baremetal-ext-status",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "BareMetalNodes",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-baremetal-nodes",
            "description": "Admin-only bare-metal node administration."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "BlockDeviceMapping",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-block-device-mapping",
            "description": "Block device mapping boot support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "BlockDeviceMappingV2Boot",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-block-device-mapping-v2-boot",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "CellCapacities",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-cell-capacities",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Cells",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "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    "
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Certificates",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-certificates",
            "description": "Certificates support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Cloudpipe",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "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    "
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "CloudpipeUpdate",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-cloudpipe-update",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ConfigDrive",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-config-drive",
            "description": "Config Drive Extension."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ConsoleAuthTokens",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-console-auth-tokens",
            "description": "Console token authentication support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ConsoleOutput",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-console-output",
            "description": "Console log output support, with tailing ability."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Consoles",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-consoles",
            "description": "Interactive Console support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "CreateBackup",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-create-backup",
            "description": "Create a backup of a server."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Createserverext",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-create-server-ext",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "DeferredDelete",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-deferred-delete",
            "description": "Instance deferred delete."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Evacuate",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-evacuate",
            "description": "Enables server evacuation."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedEvacuateFindHost",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-extended-evacuate-find-host",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedFloatingIps",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-extended-floating-ips",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedHypervisors",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-extended-hypervisors",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedNetworks",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-extended-networks",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedQuotas",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-extended-quotas",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedRescueWithImage",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-extended-rescue-with-image",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedServices",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-extended-services",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedServicesDelete",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-extended-services-delete",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedStatus",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-extended-status",
            "description": "Extended Status support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ExtendedVolumes",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-extended-volumes",
            "description": "Extended Volumes support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FixedIPs",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-fixed-ips",
            "description": "Fixed IPs support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FlavorAccess",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-flavor-access",
            "description": "Flavor access support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FlavorExtraSpecs",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-flavor-extra-specs",
            "description": "Flavors extra specs support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FlavorManage",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-flavor-manage",
            "description": "Flavor create/delete API support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FlavorRxtx",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-flavor-rxtx",
            "description": "Support to show the rxtx status of a flavor."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FlavorSwap",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-flavor-swap",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FloatingIpDns",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-floating-ip-dns",
            "description": "Floating IP DNS support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FloatingIpPools",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-floating-ip-pools",
            "description": "Floating IPs support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FloatingIps",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-floating-ips",
            "description": "Floating IPs support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "FloatingIpsBulk",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-floating-ips-bulk",
            "description": "Bulk handling of Floating IPs."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Fping",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-fping",
            "description": "Fping Management Extension."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "HideServerAddresses",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-hide-server-addresses",
            "description": "Support hiding server addresses in certain states.
"
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Hosts",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-hosts",
            "description": "Admin-only host administration."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "HypervisorStatus",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-hypervisor-status",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Hypervisors",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-hypervisors",
            "description": "Admin-only hypervisor administration."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "InstanceActions",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-instance-actions",
            "description": "View a log of actions and events taken on an
instance."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "OSInstanceUsageAuditLog",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-instance_usage_audit_log",
            "description": "Admin-only Task Log Monitoring."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Keypairs",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-keypairs",
            "description": "Keypair Support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "LockServer",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-lock-server",
            "description": "Enable lock/unlock server actions."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "MigrateServer",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-migrate-server",
            "description": "Enable migrate and live-migrate server actions."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Migrations",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-migrations",
            "description": "Provide data on migrations."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "MultipleCreate",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-multiple-create",
            "description": "Allow multiple create in the Create Server v2.1
API."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Networks",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-networks",
            "description": "Admin-only Network Management Extension."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "NetworkAssociationSupport",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-networks-associate",
            "description": "Network association support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "PauseServer",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-pause-server",
            "description": "Enable pause/unpause server actions."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Personality",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-personality",
            "description": "Personality support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "PreserveEphemeralOnRebuild",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-preserve-ephemeral-rebuild",
            "description": "Allow preservation of the ephemeral partition on
rebuild."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "QuotaClasses",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-quota-class-sets",
            "description": "Quota classes management support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Quotas",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-quota-sets",
            "description": "Quotas management support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Rescue",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-rescue",
            "description": "Instance rescue mode."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "SecurityGroupDefaultRules",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-security-group-default-rules",
            "description": "Default rules for security group support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "SecurityGroups",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-security-groups",
            "description": "Security group support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ServerDiagnostics",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-server-diagnostics",
            "description": "Allow Admins to view server diagnostics through
server action."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ServerExternalEvents",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-server-external-events",
            "description": "Server External Event Triggers."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ServerGroupQuotas",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-server-group-quotas",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ServerGroups",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-server-groups",
            "description": "Server group support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ServerListMultiStatus",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-server-list-multi-status",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ServerPassword",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-server-password",
            "description": "Server password support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ServerSortKeys",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-server-sort-keys",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "ServerStartStop",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-server-start-stop",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Services",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-services",
            "description": "Services support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Shelve",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-shelve",
            "description": "Instance shelve mode."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "SimpleTenantUsage",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-simple-tenant-usage",
            "description": "Simple tenant usage extension."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "SuspendServer",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-suspend-server",
            "description": "Enable suspend/resume server actions."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "OSTenantNetworks",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-tenant-networks",
            "description": "Tenant-based Network Management Extension."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "UsedLimits",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-used-limits",
            "description": "Provide data on limited resources that are being
 used."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "UsedLimitsForAdmin",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-used-limits-for-admin",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "UserData",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-user-data",
            "description": "Add user_data to the Create Server API."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "UserQuotas",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-user-quotas",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "VirtualInterfaces",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-virtual-interfaces",
            "description": "Virtual interface support."
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "VolumeAttachmentUpdate",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-volume-attachment-update",
            "description": ""
        },
        {
            "updated": "2014-12-03T00:00:00Z",
            "name": "Volumes",
            "links": [],
            "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
            "alias": "os-volumes",
            "description": "Volumes support."
        }
    ]
}
```

1.19.2. Show extension details

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

Normal response codes: 200203

1.19.2.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{alias}`	String	An alias for the extension name. For example, `os-server-external-events`.

This operation does not accept a request body.

1.19.2.2. Response

Example 1.104. Show extension details: JSON response

```
{
    "extension": {
        "updated": "2014-12-03T00:00:00Z",
        "name": "ServerExternalEvents",
        "links": [],
        "namespace": "http://docs.openstack.org/compute/ext/fake_xml",
        "alias": "os-server-external-events",
        "description": "Server External Event Triggers."
    }
}
```

1.20. Images

Lists, shows details for, and deletes images. Also sets, lists, shows details for, and deletes image metadata.

An image is a collection of files that you use to create and rebuild a server. By default, operators provide pre-built operating system images. You can also create custom images. See Compute server actions.

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

Method	URI	Description
GET	`/v2.1/{tenant_id}/images{?changes-since,server,name,status,minDisk,minRam,type,limit,marker}`	Lists IDs, names, and links for available images.
GET	`/v2.1/{tenant_id}/images/detail{?changes-since,server,name,status,minDisk,minRam,type,limit,marker}`	Lists all details for available images.

Method	URI	Description
GET	`/v2.1/{tenant_id}/images/{image_id}`	Shows details for an image.
DELETE	`/v2.1/{tenant_id}/images/{image_id}`	Deletes an image.

1.20.1. List images

Method	URI	Description
GET	`/v2.1/{tenant_id}/images{?changes-since,server,name,status,minDisk,minRam,type,limit,marker}`	Lists IDs, names, and links for available images.

Normal response codes: 200203

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

1.20.1.1. Request

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

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

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

Name	Type	Description
`changes-since`	DateTime *(Optional)*	Filters the response by a date and time when the image last changed status. Use this query parameter to check for changes since a previous request rather than re-downloading and re-parsing the full status at each polling interval. If data has changed, the call returns only the items changed since the `changes-since` time. If data has not changed since the `changes-since` time, the call returns an empty list. To enable you to keep track of changes, this filter also displays images that were deleted if the `changes-since` value specifies a date in the last 30 days. Items deleted more than 30 days ago might be returned, but it is not guaranteed. The date and time stamp format is ISO 8601: `CCYY-MM-DDThh:mm:ss±hh:mm` The `±hh:mm` value, if included, returns the time zone as an offset from UTC. For example, `2015-08-27T09:49:58-05:00`. If you omit the time zone, the UTC time zone is assumed.
`server`	AnyURI *(Optional)*	Filters the response by a server, as a URL.
`name`	String *(Optional)*	Filters the response by an image name, as a string.
`status`	ImageStatus *(Optional)*	Filters the response by an image status, as a string. For example, `ACTIVE`.
`minDisk`	Int *(Optional)*	Filters the response by a minimum disk size. For example, `100`.
`minRam`	Int	Filters the response by a minimum RAM size. For example, `512`.

Name	Type	Description
	(Optional)	
`type`	String *(Optional)*	Filters the response by an image type. For example, `snapshot` or `backup`. Possible values: snapshot, backup. Default: ALL.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.20.1.2. Response

Example 1.105. List images: JSON response

```
{
    "images": [
        {
            "id": "70a599e0-31e7-49b7-b260-868f441e862b",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "name": "fakeimage7"
        },
        {
            "id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "name": "fakeimage123456"
        },
        {
            "id": "a2459075-d96c-40d5-893e-577ff92e721c",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
a2459075-d96c-40d5-893e-577ff92e721c",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/a2459075-
d96c-40d5-893e-577ff92e721c",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
a2459075-d96c-40d5-893e-577ff92e721c",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "name": "fakeimage123456"
        },
        {
            "id": "a440c04b-79fa-479c-bed1-0b816eaec379",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
a440c04b-79fa-479c-bed1-0b816eaec379",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/
a440c04b-79fa-479c-bed1-0b816eaec379",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
a440c04b-79fa-479c-bed1-0b816eaec379",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "name": "fakeimage6"
        },
        {
            "id": "c905cedb-7281-47e4-8a62-f26bc5fc4c77",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "name": "fakeimage123456"
        },
        {
            "id": "cedef40a-ed67-4d10-800e-17455edce175",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
cedef40a-ed67-4d10-800e-17455edce175",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/cedef40a-
ed67-4d10-800e-17455edce175",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
cedef40a-ed67-4d10-800e-17455edce175",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "name": "fakeimage123456"
        },
        {
            "id": "76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/76fa36fc-
c930-4bf3-8c8a-ea2a2420deb6",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "name": "fakeimage123456"
        }
    ]
}
```

This operation does not return a response body.

1.20.2. List images details

Method	URI	Description
GET	`/v2.1/{tenant_id}/images/detail{?changes-since,server,name,status,minDisk,minRam,type,limit,marker}`	Lists all details for available images.

Normal response codes: 200203

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

1.20.2.1. Request

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

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

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

Name	Type	Description
`changes-since`	DateTime *(Optional)*	Filters the response by a date and time when the image last changed status. Use this query parameter to check for changes since a previous request rather than re-downloading and re-parsing the full status at each polling interval. If data has changed, the call returns only the items changed since the `changes-since` time. If data has not changed since the `changes-since` time, the call returns an empty list. To enable you to keep track of changes, this filter also displays images that were deleted if the `changes-since` value specifies a date in the last 30 days. Items deleted more than 30 days ago might be returned, but it is not guaranteed. The date and time stamp format is ISO 8601: `CCYY-MM-DDThh:mm:ss±hh:mm` The `±hh:mm` value, if included, returns the time zone as an offset from UTC. For example, `2015-08-27T09:49:58-05:00`. If you omit the time zone, the UTC time zone is assumed.
`server`	AnyURI *(Optional)*	Filters the response by a server, as a URL.
`name`	String *(Optional)*	Filters the response by an image name, as a string.
`status`	ImageStatus *(Optional)*	Filters the response by an image status, as a string. For example, `ACTIVE`.
`minDisk`	Int *(Optional)*	Filters the response by a minimum disk size. For example, `100`.
`minRam`	Int	Filters the response by a minimum RAM size. For example, `512`.

Name	Type	Description
	(Optional)	
`type`	String *(Optional)*	Filters the response by an image type. For example, `snapshot` or `backup`. Possible values: snapshot, backup. Default: ALL.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.20.2.2. Response

Example 1.106. List images details: JSON response

```
{
    "images": [
        {
            "OS-EXT-IMG-SIZE:size": "74185822",
            "created": "2011-01-01T01:02:03Z",
            "id": "70a599e0-31e7-49b7-b260-868f441e862b",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "metadata": {
                "architecture": "x86_64",
                "auto_disk_config": "True",
                "kernel_id": "nokernel",
                "ramdisk_id": "nokernel"
            },
            "minDisk": 0,
            "minRam": 0,
            "name": "fakeimage7",
            "progress": 100,
            "status": "ACTIVE",
            "updated": "2011-01-01T01:02:03Z"
        },
        {
            "OS-EXT-IMG-SIZE:size": "74185821",
            "created": "2011-01-01T01:02:03Z",
            "id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "metadata": {
                "architecture": "x86_64",
                "kernel_id": "nokernel",
                "ramdisk_id": "nokernel"
            },
            "minDisk": 0,
            "minRam": 0,
            "name": "fakeimage123456",
            "progress": 100,
            "status": "ACTIVE",
            "updated": "2011-01-01T01:02:03Z"
        },
        {
            "created": "2011-01-01T01:02:03Z",
            "id": "a2459075-d96c-40d5-893e-577ff92e721c",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
a2459075-d96c-40d5-893e-577ff92e721c",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/a2459075-
d96c-40d5-893e-577ff92e721c",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
a2459075-d96c-40d5-893e-577ff92e721c",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "metadata": {
                "kernel_id": "nokernel",
                "ramdisk_id": "nokernel"
            },
            "minDisk": 0,
            "minRam": 0,
            "name": "fakeimage123456",
            "progress": 100,
            "status": "ACTIVE",
            "updated": "2011-01-01T01:02:03Z"
        },
        {
            "created": "2011-01-01T01:02:03Z",
            "id": "a440c04b-79fa-479c-bed1-0b816eaec379",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
a440c04b-79fa-479c-bed1-0b816eaec379",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/
a440c04b-79fa-479c-bed1-0b816eaec379",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
a440c04b-79fa-479c-bed1-0b816eaec379",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "metadata": {
                "architecture": "x86_64",
                "auto_disk_config": "False",
                "kernel_id": "nokernel",
                "ramdisk_id": "nokernel"
            },
            "minDisk": 0,
            "minRam": 0,
            "name": "fakeimage6",
            "progress": 100,
            "status": "ACTIVE",
            "updated": "2011-01-01T01:02:03Z"
        },
        {
            "created": "2011-01-01T01:02:03Z",
            "id": "c905cedb-7281-47e4-8a62-f26bc5fc4c77",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "metadata": {
                "kernel_id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
                "ramdisk_id": null
            },
            "minDisk": 0,
            "minRam": 0,
            "name": "fakeimage123456",
            "progress": 100,
            "status": "ACTIVE",
            "updated": "2011-01-01T01:02:03Z"
        },
        {
            "created": "2011-01-01T01:02:03Z",
            "id": "cedef40a-ed67-4d10-800e-17455edce175",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
cedef40a-ed67-4d10-800e-17455edce175",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/cedef40a-
ed67-4d10-800e-17455edce175",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
cedef40a-ed67-4d10-800e-17455edce175",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "metadata": {
                "kernel_id": "nokernel",
                "ramdisk_id": "nokernel"
            },
            "minDisk": 0,
            "minRam": 0,
            "name": "fakeimage123456",
            "progress": 100,
            "status": "ACTIVE",
            "updated": "2011-01-01T01:02:03Z"
        },
        {
            "created": "2011-01-01T01:02:03Z",
            "id": "76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
            "links": [
                {
                    "href": "http://openstack.example.com/v2.1/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                    "rel": "self"
                },
                {
                    "href": "http://openstack.example.com/images/76fa36fc-
c930-4bf3-8c8a-ea2a2420deb6",
                    "rel": "bookmark"
                },
                {
                    "href": "http://glance.openstack.example.com/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                    "rel": "alternate",
                    "type": "application/vnd.openstack.image"
                }
            ],
            "metadata": {
                "kernel_id": "nokernel",
                "ramdisk_id": "nokernel"
            },
            "minDisk": 0,
            "minRam": 0,
            "name": "fakeimage123456",
            "progress": 100,
            "status": "ACTIVE",
            "updated": "2011-01-01T01:02:03Z"
        }
    ]
}
```

This operation does not return a response body.

1.20.3. Show image details

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

Normal response codes: 200203

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

1.20.3.1. Request

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

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

This operation does not accept a request body.

1.20.3.2. Response

Example 1.107. Show image details: JSON response

```
{
    "image": {
        "OS-EXT-IMG-SIZE:size": "74185822",
        "created": "2011-01-01T01:02:03Z",
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
            {
                "href": "http://openstack.example.com/v2.1/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                "rel": "alternate",
                "type": "application/vnd.openstack.image"
            }
        ],
        "metadata": {
            "architecture": "x86_64",
            "auto_disk_config": "True",
            "kernel_id": "nokernel",
            "ramdisk_id": "nokernel"
        },
        "minDisk": 0,
        "minRam": 0,
        "name": "fakeimage7",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    }
}
```

1.20.4. Delete image

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/images/{image_id}`	Deletes an image.

Normal response codes: 204

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

1.20.4.1. Request

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

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

This operation does not accept a request body.

1.21. Image metadata

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

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

1.21.1. Show image metadata

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

Normal response codes: 200203

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

1.21.1.1. Request

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

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

This operation does not accept a request body.

1.21.1.2. Response

Example 1.108. Show image metadata: JSON response

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

This operation does not return a response body.

1.21.2. Create or replace image metadata

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

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

Normal response codes: 200

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

1.21.2.1. Request

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

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

Example 1.109. Create or replace image metadata: JSON request

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

1.21.2.2. Response

Example 1.110. Create or replace image metadata: JSON response

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

1.21.3. Update image metadata items

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

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

Normal response codes: 200

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

1.21.3.1. Request

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

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

Example 1.111. Update image metadata items: JSON request

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

1.21.3.2. Response

Example 1.112. Update image metadata items: JSON response

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

This operation does not return a response body.

1.21.4. Show image metadata item details

Method	URI	Description
GET	`/v2.1/{tenant_id}/images/{image_id}/metadata/{key}`	Shows details for a metadata item, by key, for an image.

Normal response codes: 200203

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

1.21.4.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{image_id}`	UUID	The UUID of the image.
`{key}`	String	The metadata item key, as a string. Maximum length is 255 characters.

This operation does not accept a request body.

1.21.4.2. Response

Example 1.113. Show image metadata item details: JSON response

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

1.21.5. Create or update image metadata item

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

Normal response codes: 200

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

1.21.5.1. Request

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

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.
{image_id}	UUID	The UUID of the image.
{key}	String	The metadata item key, as a string. Maximum length is 255 characters.

Example 1.114. Create or update image metadata item: JSON request

```
{
    "meta": {
        "auto_disk_config": "False"
    }
}
```

1.21.5.2. Response

Example 1.115. Create or update image metadata item: JSON response

```
{
    "meta": {
        "auto_disk_config": "False"
    }
}
```

1.21.6. Delete image metadata item

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/images/{image_id}/metadata/{key}`	Deletes a metadata item, by key, for an image.

Normal response codes: 204

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

1.21.6.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{image_id}`	UUID	The UUID of the image.
`{key}`	String	The metadata item key, as a string. Maximum length is 255 characters.

This operation does not accept a request body.

1.22. Guest agents (os-agents)

Creates, lists, updates, and deletes guest agent builds. Use guest agents to access files on the disk, configure networking, or run other applications or scripts in the guest while the agent runs. This hypervisor-specific extension is not currently enabled for KVM. Use of guest agents is possible only if the underlying service provider uses the Xen driver.

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-agents`	Creates an agent build.
GET	`/v2.1/{tenant_id}/os-agents`	Lists agent builds.
DELETE	`/v2.1/{tenant_id}/os-agents`	Deletes an existing agent build.
PUT	`/v2.1/{tenant_id}/os-agents/{id}`	Updates an agent build.

1.22.1. Create agent build

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

Normal response codes: 201

1.22.1.1. Request

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

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

Example 1.116. Create agent build: JSON request

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

1.22.1.2. Response

Example 1.117. Create agent build: JSON response

```
{
    "agent": {
        "agent_id": 1,
        "architecture": "x86",
        "hypervisor": "hypervisor",
        "md5hash": "add6bb58e139be103324d04d82d8f545",
        "os": "os",
        "url": "http://example.com/path/to/resource",
        "version": "8.0"
    }
}
```

1.22.2. List agent builds

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

Normal response codes: 200

1.22.2.1. Request

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

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

This operation does not accept a request body.

1.22.2.2. Response

Example 1.118. List agent builds: JSON response

```
{
    "agents": [
        {
            "agent_id": 1,
            "architecture": "x86",
            "hypervisor": "hypervisor",
            "md5hash": "add6bb58e139be103324d04d82d8f545",
            "os": "os",
            "url": "http://example.com/path/to/resource",
            "version": "8.0"
        }
    ]
}
```

1.22.3. Delete agent build

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

Normal response codes: 202

1.22.3.1. Request

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

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

This operation does not accept a request body.

1.22.4. Update agent build

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

Normal response codes: 200

1.22.4.1. Request

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

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

Example 1.119. Update agent build: JSON request

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

1.22.4.2. Response

Example 1.120. Update agent build: JSON response

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

1.23. Host aggregates (os-aggregates, action)

Creates and manages host aggregates. An aggregate assigns metadata to groups of compute nodes. Aggregates are only visible to the cloud provider.

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-aggregates`	Creates an aggregate in an availability zone.
GET	`/v2.1/{tenant_id}/os-aggregates`	Lists all aggregates. Includes the ID, name, and availability zone for each aggregate.
GET	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}`	Shows the details of an aggregate, including hosts and metadata.
PUT	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}`	Updates either or both the name and availability zone for an aggregate.

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action`	Adds a host to an aggregate.
POST	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action`	Creates or replaces metadata for an aggregate.
POST	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action`	Removes a host from an aggregate.

1.23.1. Create aggregate

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

Normal response codes: 200

1.23.1.1. Request

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

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

Example 1.121. Create aggregate: JSON request

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

1.23.1.2. Response

Example 1.122. Create aggregate: JSON response

```
{
    "aggregate": {
        "availability_zone": "nova",
        "created_at": "2013-08-18T12:17:55.751757",
        "deleted": false,
        "deleted_at": null,
        "id": 1,
        "name": "name",
        "updated_at": null
    }
}
```

1.23.2. List aggregates

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-aggregates`	Lists all aggregates. Includes the ID, name, and availability zone for each aggregate.

Normal response codes: 200

1.23.2.1. Request

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

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

Example 1.123. List aggregates: JSON request

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

1.23.2.2. Response

Example 1.124. List aggregates: JSON response

```
{
    "aggregates": [
        {
            "availability_zone": "nova",
            "created_at": "2013-08-18T12:17:56.856455",
            "deleted": false,
            "deleted_at": null,
            "hosts": [],
            "id": 1,
            "metadata": {
                "availability_zone": "nova"
            },
            "name": "name",
            "updated_at": null
        }
    ]
}
```

1.23.3. Show aggregate details

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

Normal response codes: 200

1.23.3.1. Request

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

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

Example 1.125. Show aggregate details: JSON request

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

1.23.3.2. Response

Example 1.126. Show aggregate details: JSON response

```
{
    "aggregate": {
        "availability_zone": "nova",
        "created_at": "2013-08-18T12:17:56.380226",
        "deleted": false,
        "deleted_at": null,
        "hosts": [],
        "id": 1,
        "metadata": {
            "availability_zone": "nova"
        },
        "name": "name",
        "updated_at": null
    }
}
```

1.23.4. Update aggregate

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-aggregates/{aggregate_id}	Updates either or both the name and availability zone for an aggregate.

Normal response codes: 200

1.23.4.1. Request

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

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

Example 1.127. Update aggregate: JSON request

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

1.23.4.2. Response

Example 1.128. Update aggregate: JSON response

```
{
    "aggregate": {
        "availability_zone": "nova2",
        "created_at": "2013-08-18T12:17:56.259751",
        "deleted": false,
        "deleted_at": null,
        "hosts": [],
        "id": 1,
        "metadata": {
            "availability_zone": "nova2"
        },
        "name": "newname",
        "updated_at": "2013-08-18T12:17:56.286720"
    }
}
```

1.23.5. Add host

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action`	Adds a host to an aggregate.

Specify the `add_host` action in the request body.

Normal response codes: 200

1.23.5.1. Request

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

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

Example 1.129. Add host: JSON request

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

1.23.5.2. Response

Example 1.130. Add host: JSON response

```
{
    "aggregate": {
        "availability_zone": "nova",
        "created_at": "2013-08-18T12:17:56.297823",
        "deleted": false,
        "deleted_at": null,
        "hosts": [
            "21549b2f665945baaa7101926a00143c"
        ],
        "id": 1,
        "metadata": {
            "availability_zone": "nova"
        },
        "name": "name",
        "updated_at": null
    }
}
```

1.23.6. Create or update aggregate metadata

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action`	Creates or replaces metadata for an aggregate.

Specify the `add_metadata` action in the request body.

Normal response codes: 200

1.23.6.1. Request

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

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

Example 1.131. Create or update aggregate metadata: JSON request

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

1.23.6.2. Response

Example 1.132. Create or update aggregate metadata: JSON response

```
{
    "aggregate": {
        "availability_zone": "nova",
        "created_at": "2013-08-18T12:17:55.959571",
        "deleted": false,
        "deleted_at": null,
        "hosts": [],
        "id": 1,
        "metadata": {
            "availability_zone": "nova",
            "key": "value"
        },
        "name": "name",
        "updated_at": null
    }
}
```

1.23.7. Remove host

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-aggregates/{aggregate_id}/action`	Removes a host from an aggregate.

Specify the `remove_host` action in the request body.

Normal response codes: 200

1.23.7.1. Request

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

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

Example 1.133. Remove host: JSON request

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

1.23.7.2. Response

Example 1.134. Remove host: JSON response

```
{
    "aggregate": {
        "availability_zone": "nova",
        "created_at": "2013-08-18T12:17:56.990581",
        "deleted": false,
        "deleted_at": null,
        "hosts": [],
        "id": 1,
        "metadata": {
            "availability_zone": "nova"
        },
        "name": "name",
        "updated_at": null
    }
}
```

1.24. Assisted volume snapshots (os-assisted-volume-snapshots)

Creates and deletes snapshots through an emulator/hypervisor. The qcow2 file format is supported.

An internal snapshot that lacks storage such as NFS or GlusterFS can use an emulator/hypervisor to add the snapshot feature.

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-assisted-volume-snapshots`	Creates an assisted volume snapshot.
DELETE	`/v2.1/{tenant_id}/os-assisted-volume-snapshots/{snapshot_id}{?delete_info}`	Deletes an assisted volume snapshot.

1.24.1. Create assisted volume snapshots

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-assisted-volume-snapshots`	Creates an assisted volume snapshot.

Normal response codes: 200

1.24.1.1. Request

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

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

Example 1.135. Create assisted volume snapshots: JSON request

```
{
    "snapshot": {
        "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
        "create_info": {
            "snapshot_id": "421752a6-acf6-4b2d-bc7a-119f9148cd8c",
            "type": "qcow2",
            "new_file": "new_file_name"
        }
    }
}
```

1.24.1.2. Response

Example 1.136. Create assisted volume snapshots: JSON response

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

1.24.2. Delete assisted volume snapshot

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-assisted-volume-snapshots/{snapshot_id}{?delete_info}`	Deletes an assisted volume snapshot.

To make this request, add the `delete_info` query parameter to the URI, as follows:

```
DELETE /os-assisted-volume-snapshots?delete_info='{"volume_id": "521752a6-
acf6-4b2d-bc7a-119f9148cd8c"}'
```

Normal response codes: 204

1.24.2.1. Request

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

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

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

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

This operation does not accept a request body.

1.25. Availability zones (os-availability-zone)

Shows availability zone information.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-availability-zone`	Gets availability zone information.
GET	`/v2.1/{tenant_id}/os-availability-zone/detail`	Gets detailed availability zone information.

1.25.1. Get availability zone information

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-availability-zone`	Gets availability zone information.

Normal response codes: 200

1.25.1.1. Request

This table shows the URI parameters for the get availability zone information request:

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

This operation does not accept a request body.

1.25.1.2. Response

Example 1.137. Get availability zone information: JSON response

```
{
    "availabilityZoneInfo": [
        {
            "zoneState": {
                "available": true
            },
            "hosts": null,
            "zoneName": "nova"
        }
    ]
}
```

1.25.2. Get detailed availability zone information

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-availability-zone/detail`	Gets detailed availability zone information.

Normal response codes: 200

1.25.2.1. Request

This table shows the URI parameters for the get detailed availability zone information request:

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

This operation does not accept a request body.

1.25.2.2. Response

Example 1.138. Get detailed availability zone information: JSON response

```
{
    "availabilityZoneInfo": [
        {
            "zoneState": {
                "available": true
            },
            "hosts": {
                "test-host": {
                    "nova-conductor": {
                        "available": true,
                        "active": true,
                        "updated_at": "2015-04-16T08:58:55.000000"
                    },
                    "nova-cert": {
                        "available": true,
                        "active": true,
                        "updated_at": "2015-04-16T08:58:55.000000"
                    },
                    "nova-consoleauth": {
                        "available": true,
                        "active": true,
                        "updated_at": "2015-04-16T08:58:55.000000"
                    },
                    "nova-scheduler": {
                        "available": true,
                        "active": true,
                        "updated_at": "2015-04-16T08:58:55.000000"
                    },
                    "nova-network": {
                        "available": true,
                        "active": true,
                        "updated_at": "2015-04-16T08:58:54.000000"
                    }
                }
            },
            "zoneName": "internal"
        },
        {
            "zoneState": {
                "available": true
            },
            "hosts": {
                "test-host": {
                    "nova-compute": {
                        "available": true,
                        "active": true,
                        "updated_at": "2015-04-16T08:58:56.000000"
                    }
                }
            },
            "zoneName": "nova"
        }
    ]
}
```

1.26. Bare metal nodes (os-baremetal-nodes)

Bare metal nodes.

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes`	Adds a bare metal node to a server.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes`	Lists the bare metal nodes that are associated with a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action`	Adds an interface to a bare metal node that is associated with a server.
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action`	Deletes an interface from a bare metal node that is associated with a server.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}`	Shows details for a bare metal node.
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}`	Deletes a bare metal node from a server.

1.26.1. Add bare metal node

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes`	Adds a bare metal node to a server.

Preconditions

- You can add a bare metal node to a server with an `ACTIVE`, `PAUSED`, `SHUTOFF`, `VERIFY_RESIZE`, or `SOFT_DELETED` status.
- You can add a bare metal node to a server with a status that is not locked.

Normal response codes: 202

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

1.26.1.1. Request

This table shows the URI parameters for the add bare metal node request:

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

Example 1.139. Add bare metal node: JSON request

```
{
    "node": {
        "service_host": "host",
        "cpus": 8,
        "memory_mb": 8192,
        "local_gb": 128,
        "pm_address": "10.1.2.3",
        "pm_user": "pm_user",
        "pm_password": "pm_pass",
        "terminal_port": 8000
    }
}
```

Example 1.140. Add bare metal node: JSON request

```
{
    "node": {
        "service_host": "host",
        "cpus": 8,
        "memory_mb": 8192,
        "local_gb": 128,
        "pm_address": "10.1.2.3",
        "pm_user": "pm_user",
        "pm_password": "pm_pass",
        "prov_mac_address": "12:34:56:78:90:ab",
        "terminal_port": 8000
    }
}
```

1.26.1.2. Response

Example 1.141. Add bare metal node: JSON response

```
{
    "node": {
        "cpus": 8,
        "id": 1,
        "instance_uuid": null,
        "interfaces": [],
        "local_gb": 128,
        "memory_mb": 8192,
        "pm_address": "10.1.2.3",
        "pm_user": "pm_user",
        "pxe_config_path": null,
        "service_host": "host",
        "task_state": null,
        "terminal_port": 8000,
        "updated_at": null,
        "uuid": "73d35253-b6fb-4c83-b8eb-0229336e79b6"
    }
}
```

Example 1.142. Add bare metal node: JSON response

```
{
    "node": {
        "cpus": 8,
        "id": 1,
        "instance_uuid": null,
        "interfaces": [
            {
                "address": "12:34:56:78:90:ab",
                "datapath_id": null,
                "id": 1,
                "port_no": null
            }
        ],
        "local_gb": 128,
        "memory_mb": 8192,
        "pm_address": "10.1.2.3",
        "pm_user": "pm_user",
        "pxe_config_path": null,
        "service_host": "host",
        "task_state": null,
        "terminal_port": 8000,
        "updated_at": null,
        "uuid": "0a130464-bccc-4e36-b9d3-9a8c98e636ae"
    }
}
```

1.26.2. List bare metal nodes

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes`	Lists the bare metal nodes that are associated with a server.

Normal response codes: 202

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

1.26.2.1. Request

This table shows the URI parameters for the list bare metal nodes request:

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

This operation does not accept a request body.

1.26.2.2. Response

Example 1.143. List bare metal nodes: JSON response

```
{
    "nodes": [
        {
            "cpus": 8,
            "id": 1,
            "instance_uuid": null,
            "interfaces": [
                {
                    "address": "aa:aa:aa:aa:aa:aa",
                    "datapath_id": null,
                    "id": 1,
                    "port_no": null
                }
            ],
            "local_gb": 128,
            "memory_mb": 8192,
            "pm_address": "10.1.2.3",
            "pm_user": "pm_user",
            "pxe_config_path": null,
            "service_host": "host",
            "task_state": null,
            "terminal_port": 8000,
            "updated_at": null,
            "uuid": "6fae68da-108b-4a9d-87c4-88831ee1241b"
        }
    ]
}
```

1.26.3. Add interface to bare metal node

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action`	Adds an interface to a bare metal node that is associated with a server.

Normal response codes: 202

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

1.26.3.1. Request

This table shows the URI parameters for the add interface to bare metal node request:

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

Example 1.144. Add interface to bare metal node: JSON request

```
{
    "add_interface": {
        "address": "aa:aa:aa:aa:aa:aa"
    }
}
```

1.26.3.2. Response

Example 1.145. Add interface to bare metal node: JSON response

```
{
    "interface": {
        "address": "aa:aa:aa:aa:aa:aa",
        "datapath_id": null,
        "id": 1,
        "port_no": null
    }
}
```

1.26.4. Delete interface from bare metal node

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/action`	Deletes an interface from a bare metal node that is associated with a server.

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

1.26.4.1. Request

This table shows the URI parameters for the delete interface from bare metal node request:

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

Example 1.146. Delete interface from bare metal node: JSON request

```
{
    "remove_interface": {
        "address": "aa:aa:aa:aa:aa:aa"
    }
}
```

1.26.5. Show bare metal node details

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}`	Shows details for a bare metal node.

Preconditions

- The bare metal node must be associated with the server.

Normal response codes: 202

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

1.26.5.1. Request

This table shows the URI parameters for the show bare metal node details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{server_id}`	UUID	The UUID of the server.
`{node_id}`	String	The node ID.

This operation does not accept a request body.

1.26.5.2. Response

Example 1.147. Show bare metal node details: JSON response

```
{
    "node": {
        "cpus": 8,
        "id": 1,
        "instance_uuid": null,
        "interfaces": [
            {
                "address": "aa:aa:aa:aa:aa:aa",
                "datapath_id": null,
                "id": 1,
                "port_no": null
            }
        ],
        "local_gb": 128,
        "memory_mb": 8192,
        "pm_address": "10.1.2.3",
        "pm_user": "pm_user",
        "pxe_config_path": null,
        "service_host": "host",
        "task_state": null,
        "terminal_port": 8000,
        "updated_at": null,
        "uuid": "c862b836-c7c1-4f7f-8081-6766fa9cf38b"
    }
}
```

1.26.6. Delete bare metal node

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/os-baremetal-nodes/{node_id}`	Deletes a bare metal node from a server.

1.26.6.1. Request

This table shows the URI parameters for the delete bare metal node request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{server_id}`	UUID	The UUID of the server.
`{node_id}`	String	The node ID.

This operation does not accept a request body.

1.27. Servers with block device mapping format (servers, os-block-device-mapping)

Creates a server with a block device mapping.

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers`	Creates a server with a block device mapping.

1.27.1. Create server with block device mapping

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers`	Creates a server with a block device mapping.

To define the block device mapping, you can include either a `block_device_mapping` or `block_device_mapping_v2` object in the request body. The `block_device_mapping_v2` object is preferred.

Normal response codes: 202

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

1.27.1.1. Request

This table shows the URI parameters for the create server with block device mapping request:

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

Example 1.148. Create server with block device mapping: JSON request

```
{
    "server": {
        "name": "new-server-test",
        "imageRef": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "flavorRef": "http://openstack.example.com/openstack/flavors/1",
        "metadata": {
            "My Server Name": "Apache1"
        },
        "personality": [
            {
                "path": "/etc/banner.txt",
                "contents":
 "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcyBpbiBqdXN0IHN1Y2ggYSBkaXJlY3Rpb2
="
            }
        ],
        "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"
            }
        ]
    }
}
```

1.27.1.2. Response

Example 1.149. Create server with block device mapping: JSON response

```
{
    "server": {
        "adminPass": "N4x7wFX6iN8D",
        "id": "babd1af0-4fc6-4529-b32f-aad69811ccf5",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/servers/
babd1af0-4fc6-4529-b32f-aad69811ccf5",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
babd1af0-4fc6-4529-b32f-aad69811ccf5",
                "rel": "bookmark"
            }
        ]
    }
}
```

1.28. Cells (os-cells, capacities)

Adds neighbor cells, lists neighbor cells, and shows the capabilities of the local cell.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-cells`	Lists cells.
GET	`/v2.1/{tenant_id}/os-cells`	Lists cells with details.
GET	`/v2.1/{tenant_id}/os-cells/{cell_id}`	Shows data for a cell.
GET	`/v2.1/{tenant_id}/os-cells/{cell_id}/capacities`	Shows capacities for a cell.

1.28.1. List cells

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-cells`	Lists cells.

When cells are not enabled, the call returns the `Not Implemented (501)` response code.

Normal response codes: 200

Error response codes: notImplemented (501)

1.28.1.1. Request

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

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

This operation does not accept a request body.

1.28.1.2. Response

Example 1.150. 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"
        }
    ]
}
```

1.28.2. List cells with details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-cells`	Lists cells with details.

When cells are not enabled, the call returns the `Not Implemented (501)` response code.

Normal response codes: 200

Error response codes: notImplemented (501)

1.28.2.1. Request

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

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

This operation does not accept a request body.

1.28.2.2. Response

Example 1.151. List cells with details: JSON response

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

1.28.3. Show cell data

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-cells/{cell_id}`	Shows data for a cell.

When cells are not enabled, the call returns the `Not Implemented (501)` response code.

Normal response codes: 200

Error response codes: notImplemented (501)

1.28.3.1. Request

This table shows the URI parameters for the show cell data request:

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

This operation does not accept a request body.

1.28.3.2. Response

Example 1.152. Show cell data: JSON response

```
{
    "cell": {
        "name": "cell3",
        "rpc_host": null,
        "rpc_port": null,
        "type": "child",
        "username": "username3"
    }
}
```

1.28.4. Show cell capacities

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-cells/{cell_id}/capacities`	Shows capacities for a cell.

When cells are not enabled, the call returns the `Not Implemented (501)` response code.

Normal response codes: 200

Error response codes: notImplemented (501)

1.28.4.1. Request

This table shows the URI parameters for the show cell capacities request:

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

This operation does not accept a request body.

1.28.4.2. Response

Example 1.153. Show cell capacities: JSON response

```
{
    "cell": {
        "capacities": {
            "disk_free": {
                "total_mb": 1052672,
                "units_by_mb": {
                    "0": 0,
                    "163840": 5,
                    "20480": 46,
                    "40960": 23,
                    "81920": 11
                }
            },
            "ram_free": {
                "total_mb": 7680,
                "units_by_mb": {
                    "16384": 0,
                    "2048": 3,
                    "4096": 1,
                    "512": 13,
                    "8192": 0
                }
            }
        }
    }
}
```

1.29. Root certificates (os-certificates)

Creates and shows details for a root certificate.

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-certificates`	Creates a certificate.
GET	`/v2.1/{tenant_id}/os-certificates/{certificate_id}`	Shows details for a certificate.

1.29.1. Create certificate

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-certificates`	Creates a certificate.

Normal response codes: 200

1.29.1.1. Request

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

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

This operation does not accept a request body.

1.29.1.2. Response

Example 1.154. Create certificate: JSON response

```
{
    "certificate": {
        "data": "Certificate:\n     Data:\n        Version: 1 (0x0)\n
 Serial Number: 1018 (0x3fa)\n     Signature Algorithm: md5WithRSAEncryption
\n        Issuer: O=NOVA ROOT, L=Mountain View, ST=California, C=US\n
      Validity\n            Not Before: Aug 12 07:20:30 2013 GMT\n
       Not After : Aug 12 07:20:30 2014 GMT\n         Subject: C=US, ST=
California, O=OpenStack, OU=NovaDev, CN=openstack-fake-2013-08-12T07:20:30Z
\n        Subject Public Key Info:\n             Public Key Algorithm:
 rsaEncryption\n                Public-Key: (1024 bit)\n
 Modulus:\n                     00:ac:ff:b1:d1:ed:54:4e:35:6c:34:b4:8f:0b:04:
\n                    50:25:a3:e2:4f:02:4c:4f:26:59:bd:f3:fd:eb:da:
\n                    18:c2:36:aa:63:42:72:1f:88:4f:3a:ec:e7:9f:8e:
\n                    44:2a:d3:b8:94:7b:20:41:f8:48:02:57:91:4c:16:
\n                    62:f1:21:d4:f2:40:b5:86:50:d9:61:f0:be:ff:d8:
\n                    8d:9f:4b:aa:6a:07:38:a2:7f:87:21:fc:e6:6e:1d:
\n                    0a:95:1a:90:0e:60:c2:24:e9:8e:e8:68:1b:e9:f3:
\n                    c6:b0:7c:da:c5:20:66:9b:85:ea:f5:c9:a7:de:ee:
\n                    16:b1:51:a0:4d:e3:95:98:df\n
 Exponent: 65537 (0x10001)\n     Signature Algorithm: md5WithRSAEncryption
\n         15:42:ca:71:cc:32:af:dc:cf:45:91:df:8a:b8:30:c4:7f:78:
\n         80:a7:25:c2:d9:81:3e:b3:dd:22:cc:3b:f8:94:e7:8f:04:f6:
\n         93:04:9e:85:d4:10:40:ff:5a:07:47:24:b5:ae:93:ad:8d:e1:
\n         e6:54:4a:8d:4a:29:53:c4:8d:04:6b:0b:f6:af:38:78:02:c5:
\n         05:19:89:82:2d:ba:fd:11:3c:1e:18:c9:0c:3d:03:93:6e:bc:
\n         66:70:34:ee:03:78:8a:1d:3d:64:e8:20:2f:90:81:8e:49:1d:
\n         07:37:15:66:42:cb:58:39:ad:56:ce:ed:47:c6:78:0b:0e:75:
\n         29:ca\n-----BEGIN CERTIFICATE-----\
nMIICNDCCAZ0CAgP6MA0GCSqGSIb3DQEBBAUAME4xEjAQBgNVBAoTCU5PVkEgUk9P\
nVDEWMBQGA1UEBxMNTW91bnRhaW4gVmlldzETMBEGA1UECBMKQ2FsaWZvcm5pYTEL\
nMAkGA1UEBhMCVVMwHhcNMTMwODEyMDcyMDMwWhcNMTQwODEyMDcyMDMwWjB2MQsw\
nCQYDVQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcm5pYTESMBAGA1UECgwJT3BlblN0\
nYWNrMRAwDgYDVQQLDAdOb3ZhRGV2MSwwKgYDVQQDDCNvcGVuc3RhY2stZmFrZS0y\
nMDEzLTA4LTEyVDA3OjIwOjMwWjCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA\
nrP+x0e1UTjVsNLSPCwRQJaPiTwJMTyZZvfP969oYwjaqY0JyH4hPOuznn45EKtO4\
nlHsgQfhIAleRTBZi8SHU8kC1hlDZYfC+/9iNn0uqagc4on+HIfzmbh0KlRqQDmDC\
nJOmO6Ggb6fPGsHzaxSBmm4Xq9cmn3u4WsVGgTeOVmN8CAwEAATANBgkqhkiG9w0B\
nAQQFAAOBgQAVQspxzDKv2.1M9Fkd+KuDDEf3iApyXC2YE+s90izDv4lOePBPaTBJ6F
\n1BBA/1oHRyS1rpOtjeHmVEqNSilTxI0Eawv2rzh4AsUFGYmCLbr9ETweGMkMPQOT\
nbrxmcDTuA3iKHT1k6CAvkIGOSR0HNxVmQstYOa1Wzu1HxngLDnUpyg==\n-----END
 CERTIFICATE-----\n",
        "private_key": "-----BEGIN RSA PRIVATE KEY-----\
nMIICXgIBAAKBgQCs/7HR7VRONWw0tI8LBFAlo+JPAkxPJlm98/3r2hjCNqpjQnIf\
niE867OefjkQq07iUeyBB+EgCV5FMFmLxIdTyQLWGUNlh8L7/2I2fS6pqBziif4ch\
n/OZuHQqVGpAOYMIk6Y7oaBvp88awfNrFIGabher1yafe7haxUaBN45WY3wIDAQAB\
nAoGBAIrcr2I/KyWf0hw4Nn10V9TuyE/9Gz2JHg3QFKjFJox2DqygADT5WAeHc6Bq\
nNKNf0NA2SL1LSpm+ql01tvOw4VjE5TF6OHiIzHuTTnXggG6vuA8rxp6L24HtkAcC\
n0CBno9ggSX6jVornJPBfxpkwITYSvH57BUFVD7ovbPyWGzS5AkEA1JeUtL6zxwps\
nWRr1aJ8Ill2uQk/RUIvSZOU61s+B190zvHikFy8LD8CI6vvBmjC/IZuZVedufjqs\
n4vX82uDO3QJBANBSh2b2dyB4AGVFY9vXMRtALAspJHbLHy+zTKxlGPFiuz7Se3ps\
n8Kehz4C/CBXgQkk194dwFSGE19/PQfyJROsCQQCFFDJZhrtBUMwMZ2zSRiN5BUGt\
nbwuncS+OS1Su3Yz5VRYq2BZYEPHKtYrAFkLWQ8eRwTaWaN5pFE/fb38OgQXdAkA4\
nDm0W/K0zlHbuyUxEpNQ28/6mBi0ktiWvLT0tioq6sYmXLwZA/D2JrhXrG/xt/ol3\
nr8jqrfNRsLByLhAgh0N/AkEAl2eR0O97lTEgFNqzIQwVmIAn9mBO3cnf3tycvlDU\
nm6eb2CS242y4QalfCCAEjxoJURdfsm3/D1iFo00X+IWF+A==\n-----END RSA PRIVATE
 KEY-----\n"
    }
}
```

1.29.2. Show certificate details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-certificates/{certificate_id}`	Shows details for a certificate.

Normal response codes: 200

1.29.2.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{certificate_id}`	String	The certificate ID, as a string. A valid value is `root`.

This operation does not accept a request body.

1.29.2.2. Response

Example 1.155. Show certificate details: JSON response

```
{
    "certificate": {
        "data": "-----BEGIN CERTIFICATE-----\
nMIICyzCCAjSgAwIBAgIJAJ8zSIxUp/m4MA0GCSqGSIb3DQEBBAUAME4xEjAQBgNV\
nBAoTCU5PVkEgUk9PVDEWMBQGA1UEBxMNTW91bnRhaW4gVmlldzETMBEGA1UECBMK\
nQ2FsaWZvcm5pYTELMAkGA1UEBhMCVVMwHhcNMTIxMDE3MDEzMzM5WhcNMTMxMDE3\
nMDEzMzM5WjBOMRIwEAYDVQQKEwlOT1ZBIFJPT1QxFjAUBgNVBAcTDU1vdW50YWlu\
nIFZpZXcxEzARBgNVBAgTCkNhbGlmb3JuaWExCzAJBgNVBAYTAlVTMIGfMA0GCSqG\
nSIb3DQEBAQUAA4GNADCBiQKBgQDXW4QfQQxJG4MqurqK8nU/Lge0mfNKxXj/Gwvg\
n2sQVwxzmKfoxih8Nn6yt0yHMNjhoji1UoWI03TXUnPZRAZmsypGKZeBd7Y1ZOCPB\
nXGZVGrQm+PB2kZU+3cD8fVKcueMLLeZ+LRt5d0njnoKhc5xjqMlfFPimHMba4OL6\
nTnYzPQIDAQABo4GwMIGtMAwGA1UdEwQFMAMBAf8wHQYDVR0OBBYEFKyoKu4SMOFM\
ngx5Ec7p0nrCkabvxMH4GA1UdIwR3MHWAFKyoKu4SMOFMgx5Ec7p0nrCkabvxoVKk\
nUDBOMRIwEAYDVQQKEwlOT1ZBIFJPT1QxFjAUBgNVBAcTDU1vdW50YWluIFZpZXcx\
nEzARBgNVBAgTCkNhbGlmb3JuaWExCzAJBgNVBAYTAlVTggkAnzNIjFSn+bgwDQYJ\
nKoZIhvcNAQEEBQADgYEAXuvXlu1o/SVvykSLhHW8QiAY00yzN/eDzYmZGomgiuoO\n/
x+ayVzbrz1UWZnBD+lC4hll2iELSmf22LjLoF+s/9NyPqHxGL3FrfatBkndaiF8\nAx/
TMEyCPl7IQWi+3zzatqOKHSHiG7a9SGn/7o2aNTIWKVulfy5GvmbBjBM/0UE=\n-----END
 CERTIFICATE-----\n",
        "private_key": null
    }
}
```

1.30. Cloudpipe (os-cloudpipe)

Manages virtual VPNs for projects.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-cloudpipe`	Lists cloudpipes.
POST	`/v2.1/{tenant_id}/os-cloudpipe`	Creates a cloudpipe.

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-cloudpipe/configure-project`	Updates the virtual private network (VPN) IP address and port for a cloudpipe instance.

1.30.1. List cloudpipes

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-cloudpipe`	Lists cloudpipes.

Normal response codes: 200

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

1.30.1.1. Request

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

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

This operation does not accept a request body.

1.30.1.2. Response

Example 1.156. List cloudpipes: JSON response

```
{
    "cloudpipes": [
        {
            "created_at": "2012-11-27T17:18:01Z",
            "instance_id": "27deecdb-baa3-4a26-9c82-32994b815b01",
            "internal_ip": "192.168.0.3",
            "project_id": "fa1765bd-a352-49c7-a6b7-8ee108a3cb0c",
            "public_ip": "127.0.0.1",
            "public_port": 22,
            "state": "down"
        }
    ]
}
```

1.30.2. Create cloudpipe

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-cloudpipe`	Creates a cloudpipe.

Normal response codes: 200

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

1.30.2.1. Request

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

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

Example 1.157. Create cloudpipe: JSON request

```
{
    "cloudpipe": {
        "project_id": "059f21e3-c20e-4efc-9e7a-eba2ab3c6f9a"
    }
}
```

This operation does not accept a request body.

1.30.2.2. Response

Example 1.158. Create cloudpipe: JSON response

```
{
    "instance_id": "1e9b8425-34af-488e-b969-4d46f4a6382e"
}
```

1.30.3. Update cloudpipe

Method	URI	Description
POST	/v2.1/{tenant_id}/os-cloudpipe/configure-project	Updates the virtual private network (VPN) IP address and port for a cloudpipe instance.

Normal response codes: 202

Error response codes: computeFault (400, 500, …), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), badMediaType (415), NetworkNotFound (400)

1.30.3.1. Request

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

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

Example 1.159. Update cloudpipe: JSON request

```
{
    "configure_project": {
        "vpn_ip": "192.168.1.1",
        "vpn_port": "2000"
    }
}
```

This operation does not accept a request body.

1.31. Server consoles (servers, os-consoles, os-console-auth-token)

Manages server consoles.

Method	URI	Description
POST	/v2.1/{tenant_id}/servers/{server_id}/os-consoles	Creates a console for a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-consoles	Lists all consoles for a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-consoles/{console_id}	Shows details for a console for a server instance.
DELETE	/v2.1/{tenant_id}/servers/{server_id}/os-consoles/{console_id}	Deletes a console for a server instance.
GET	/v2.1/{tenant_id}/servers/{server_id}/os-console-auth-token	Shows the authentication token for a console for a server instance.

1.31.1. Create console

Method	URI	Description
POST	`/v2.1/{tenant_id}/servers/{server_id}/os-consoles`	Creates a console for a server instance.

Normal response codes: 200

1.31.1.1. Request

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

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

This operation does not accept a request body.

1.31.2. Lists consoles

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-consoles`	Lists all consoles for a server instance.

Normal response codes: 200

1.31.2.1. Request

This table shows the URI parameters for the lists consoles request:

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

This operation does not accept a request body.

1.31.2.2. Response

Example 1.160. List consoles: JSON response

```
{
    "consoles": [
        {
            "console": {
                "console_type": "fake",
                "id": 1
            }
        }
    ]
}
```

1.31.3. Show console details

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-consoles/{console_id}`	Shows details for a console for a server instance.

Normal response codes: 200

1.31.3.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{server_id}`	UUID	The UUID of the server.
`{console_id}`	UUID	The UUID of the console.

This operation does not accept a request body.

1.31.3.2. Response

Example 1.161. List consoles: JSON response

```
{
    "consoles": [
        {
            "console": {
                "console_type": "fake",
                "id": 1
            }
        }
    ]
}
```

1.31.4. Delete console

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/servers/{server_id}/os-consoles/{console_id}`	Deletes a console for a server instance.

Normal response codes: 202

1.31.4.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{server_id}`	UUID	The UUID of the server.
`{console_id}`	UUID	The UUID of the console.

This operation does not accept a request body.

1.31.5. Show console authentication token

Method	URI	Description
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-console-auth-token`	Shows the authentication token for a console for a server instance.

This feature is available for `rdp-html5` console type only.

Normal response codes: 200

1.31.5.1. Request

This table shows the URI parameters for the show console authentication token request:

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

This operation does not accept a request body.

1.31.5.2. Response

Example 1.162. List consoles: JSON response

```
{
    "console": {
        "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
        "host": "localhost",
        "port": 5900,
        "internal_access_path": "51af38c3-555e-4884-a314-6c8cdde37444"
    }
}
```

1.32. Fixed IPs (os-fixed-ips)

Shows data for a fixed IP, such as host name, CIDR, and address. Also, reserves and frees a fixed IP address.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}`	Shows details for a fixed IP address.
POST	`/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}/action`	Reserves or releases a fixed IP.

1.32.1. Show fixed IP details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}`	Shows details for a fixed IP address.

Normal response codes: 200

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

1.32.1.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{fixed_ip}`	String	The fixed IP of interest to you.

This operation does not accept a request body.

1.32.1.2. Response

Example 1.163. Show fixed IP details: JSON response

```
{
    "fixed_ip": {
        "address": "192.168.1.1",
        "cidr": "192.168.1.0/24",
        "host": "host",
        "hostname": "openstack"
    }
}
```

1.32.2. Reserve or release a fixed IP

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-fixed-ips/{fixed_ip}/action`	Reserves or releases a fixed IP.

To reserve a fixed IP address, specify `reserve` in the request body. To release a fixed IP address, specify `unreserve` in the request body.

Normal response codes: 202

Error response codes: computeFault (400, 500, …), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), badMediaType (415), NetworkNotFound (400)

1.32.2.1. Request

This table shows the URI parameters for the reserve or release a fixed ip request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{fixed_ip}`	String	The fixed IP of interest to you.

Example 1.164. Reserve or release a fixed IP: JSON request

```
{
    "reserve": null
}
```

1.33. Floating IP DNS records (os-floating-ip-dns)

Manages DNS records associated with floating IP addresses. The API dispatches requests to a DNS driver that is selected at startup.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ip-dns`	Lists registered DNS domains published by the DNS drivers.
PUT	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}`	Creates or updates a DNS domain.
DELETE	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}`	Deletes a DNS domain and all associated host entries.
PUT	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}`	Creates or updates a DNS entry.
GET	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}`	Finds a unique DNS entry for a domain and name.
DELETE	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}`	Deletes a DNS entry.
GET	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}`	Lists DNS entries for a domain and IP.

1.33.1. List DNS domains

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ip-dns`	Lists registered DNS domains published by the DNS drivers.

Normal response codes: 200

1.33.1.1. Request

This table shows the URI parameters for the list dns domains request:

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

This operation does not accept a request body.

1.33.1.2. Response

Example 1.165. List DNS domains: JSON response

```
{
    "domain_entries": [
        {
            "availability_zone": null,
            "domain": "domain1.example.org",
            "project": "project1",
            "scope": "public"
        }
    ]
}
```

1.33.2. Create or update DNS domain

Method	URI	Description
PUT	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}`	Creates or updates a DNS domain.

Normal response codes: 200

1.33.2.1. Request

This table shows the URI parameters for the create or update dns domain request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{domain}`	String	The registered DNS domain that the DNS drivers publish.

Example 1.166. Create or update DNS domain: JSON request

```
{
    "domain_entry": {
        "scope": "public",
        "project": "project1"
    }
}
```

1.33.2.2. Response

Example 1.167. Create or update DNS domain: JSON response

```
{
    "domain_entry": {
        "availability_zone": null,
        "domain": "domain1.example.org",
        "project": "project1",
        "scope": "public"
    }
}
```

1.33.3. Delete DNS domain

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}`	Deletes a DNS domain and all associated host entries.

Normal response codes: 200

1.33.3.1. Request

This table shows the URI parameters for the delete dns domain request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{domain}`	String	The registered DNS domain that the DNS drivers publish.

This operation does not accept a request body.

1.33.4. Create or update DNS entry

Method	URI	Description
PUT	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}`	Creates or updates a DNS entry.

Normal response codes: 200

1.33.4.1. Request

This table shows the URI parameters for the create or update dns entry request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{domain}`	String	The registered DNS domain that the DNS drivers publish.
`{name}`	String	The name of the DNS entry.

Example 1.168. Create or update DNS entry: JSON request

```
{
    "dns_entry": {
        "ip": "192.168.53.11",
        "dns_type": "A"
    }
}
```

1.33.4.2. Response

Example 1.169. Create or update DNS entry: JSON response

```
{
    "dns_entry": {
        "domain": "domain1.example.org",
        "id": null,
        "ip": "192.168.1.1",
        "name": "instance1",
        "type": "A"
    }
}
```

1.33.5. Find unique DNS entry

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}`	Finds a unique DNS entry for a domain and name.

Normal response codes: 200

1.33.5.1. Request

This table shows the URI parameters for the find unique dns entry request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{domain}`	String	The registered DNS domain that the DNS drivers publish.
`{name}`	String	The name of the DNS entry.

This operation does not accept a request body.

1.33.5.2. Response

Example 1.170. Find unique DNS entry: JSON response

```
{
    "dns_entry": {
        "domain": "domain1.example.org",
        "id": null,
        "ip": "192.168.1.1",
        "name": "instance1",
        "type": null
    }
}
```

1.33.6. Delete DNS entry

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}`	Deletes a DNS entry.

Normal response codes: 200

1.33.6.1. Request

This table shows the URI parameters for the delete dns entry request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{domain}`	String	The registered DNS domain that the DNS drivers publish.
`{name}`	String	The name of the DNS entry.

This operation does not accept a request body.

1.33.7. List DNS entries

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}`	Lists DNS entries for a domain and IP.

Normal response codes: 200

1.33.7.1. Request

This table shows the URI parameters for the list dns entries request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{domain}`	String	The registered DNS domain that the DNS drivers publish.
`{ip}`	String	The IP address.

This operation does not accept a request body.

1.33.7.2. Response

Example 1.171. List DNS entries: JSON response

```
{
    "dns_entries": [
        {
            "domain": "domain1.example.org",
            "id": null,
            "ip": "192.168.1.1",
            "name": "instance1",
            "type": null
        }
    ]
}
```

1.34. Floating IP pools (os-floating-ip-pools)

Manages groups of floating IPs.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ip-pools`	Lists floating IP pools.

1.34.1. List floating IP pools

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ip-pools`	Lists floating IP pools.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.34.1.1. Request

This table shows the URI parameters for the list floating ip pools request:

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

This operation does not accept a request body.

1.34.1.2. Response

Example 1.172. List floating IP pools: JSON response

```
{
    "floating_ip_pools": [
        {
            "name": "pool1"
        },
        {
            "name": "pool2"
        }
    ]
}
```

1.35. Floating IPs (os-floating-ips)

Lists floating IP addresses for a project. Also, creates (allocates) a floating IP address for a project, shows floating IP address details, and deletes (deallocates) a floating IP address from a project.

The cloud administrator configures a pool of floating IP addresses in OpenStack Compute. The project quota defines the maximum number of floating IP addresses that you can allocate to the project. After you allocate a floating IP address for a project, you can:

- Add (associate) the floating IP address with an instance in the project. You can associate only one floating IP address with an instance at a time.
- Remove (disassociate) the floating IP address from an instance in the project.
- Delete, or deallocate, a floating IP from the project, which automatically deletes any associations for that IP address.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ips`	Lists floating IP addresses associated with the tenant or account.
POST	`/v2.1/{tenant_id}/os-floating-ips`	Creates, or allocates, a floating IP address for the current project. By default, the floating IP address is allocated from the public pool.
GET	`/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}`	Shows details for a floating IP address, by ID, that is associated with the tenant or account.
DELETE	`/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}`	Deletes, or deallocates, a floating IP address from the current project and returns it to the pool from which it was allocated.

1.35.1. List floating IP addresses

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ips`	Lists floating IP addresses associated with the tenant or account.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.35.1.1. Request

This table shows the URI parameters for the list floating ip addresses request:

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

This operation does not accept a request body.

1.35.1.2. Response

Example 1.173. List floating IP addresses: JSON response

```
{
    "floatingips": [
        {
            "router_id": "d23abc8d-2991-4a55-ba98-2aaea84cc72f",
            "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
            "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
            "fixed_ip_address": "10.0.0.3",
            "floating_ip_address": "172.24.4.228",
            "port_id": "ce705c24-c1ef-408a-bda3-7bbd946164ab",
            "id": "2f245a7b-796b-4f26-9cf9-9e82d248fda7",
            "status": "ACTIVE"
        },
        {
            "router_id": null,
            "tenant_id": "4969c491a3c74ee4af974e6d800c62de",
            "floating_network_id": "376da547-b977-4cfe-9cba-275c80debf57",
            "fixed_ip_address": null,
            "floating_ip_address": "172.24.4.227",
            "port_id": null,
            "id": "61cea855-49cb-4846-997d-801b70c71bdd",
            "status": "DOWN"
        }
    ]
}
```

1.35.2. Create (allocate) floating IP address

Method	URI	Description
POST	/v2.1/{tenant_id}/os-floating-ips	Creates, or allocates, a floating IP address for the current project. By default, the floating IP address is allocated from the public pool.

If more than one floating IP address pool is available, use the `pool` parameter to specify from which pool to allocate the IP address.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

Error response codes: 400

1.35.2.1. Request

This table shows the URI parameters for the create (allocate) floating ip address request:

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

Example 1.174. Create (allocate) floating IP address: JSON request

```
{
    "pool": "public"
}
```

1.35.2.2. Response

Example 1.175. Create (allocate) floating IP address: JSON response

```
{
    "floating_ip": {
        "instance_id": null,
        "ip": "172.24.4.4",
        "fixed_ip": null,
        "id": "c9c04158-3ed4-449c-953a-aa21fb47cde7",
        "pool": "public"
    }
}
```

1.35.3. Show floating IP address details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}`	Shows details for a floating IP address, by ID, that is associated with the tenant or account.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.35.3.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{floating_ip_id}`	Int	The ID of the floating IP address.

This operation does not accept a request body.

1.35.3.2. Response

Example 1.176. Show floating IP address details: JSON response

```
{
    "floating_ip": {
        "instance_id": null,
        "ip": "172.24.4.3",
        "fixed_ip": null,
        "id": "b310fff3-c467-4950-9b00-038afebd151c",
        "pool": "public"
    }
}
```

1.35.4. Delete (deallocate) floating IP address

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-floating-ips/{floating_ip_id}`	Deletes, or deallocates, a floating IP address from the current project and returns it to the pool from which it was allocated.

If the IP address is still associated with a running instance, it is automatically disassociated from that instance.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.35.4.1. Request

This table shows the URI parameters for the delete (deallocate) floating ip address request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{floating_ip_id}`	Int	The ID of the floating IP address.

This operation does not accept a request body.

1.36. Floating IPs bulk (os-floating-ips-bulk)

(nova-network only) Bulk-creates, deletes, and lists floating IPs. Default pool name is `nova`.

To view available pools, use the `os-floating-ip-pools` extension.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ips-bulk`	Lists all floating IPs.
POST	`/v2.1/{tenant_id}/os-floating-ips-bulk`	Bulk-creates floating IPs.
POST	`/v2.1/{tenant_id}/os-floating-ips-bulk/delete`	Bulk-deletes floating IPs.
GET	`/v2.1/{tenant_id}/os-floating-ips-bulk/{host_name}`	Lists all floating IPs for a host.

1.36.1. List floating IPs

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ips-bulk`	Lists all floating IPs.

Normal response codes: 200

1.36.1.1. Request

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

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

This operation does not accept a request body.

1.36.1.2. Response

Example 1.177. List floating IPs: JSON response

```
{
    "floating_ip_info": [
        {
            "address": "10.10.10.1",
            "instance_uuid": null,
            "fixed_ip": null,
            "interface": "eth0",
            "pool": "nova",
            "project_id": null
        },
        {
            "address": "10.10.10.2",
            "instance_uuid": null,
            "fixed_ip": null,
            "interface": "eth0",
            "pool": "nova",
            "project_id": null
        },
        {
            "address": "10.10.10.3",
            "instance_uuid": null,
            "fixed_ip": null,
            "interface": "eth0",
            "pool": "nova",
            "project_id": null
        }
    ]
}
```

1.36.2. Create floating IPs

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-floating-ips-bulk`	Bulk-creates floating IPs.

Normal response codes: 200

1.36.2.1. Request

This table shows the URI parameters for the create floating ips request:

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

Example 1.178. Create floating IPs: JSON request

```
{
    "floating_ips_bulk_create": {
        "ip_range": "192.168.1.0/24",
        "pool": "nova",
        "interface": "eth0"
    }
}
```

1.36.2.2. Response

Example 1.179. Create floating IPs: JSON response

```
{
    "floating_ips_bulk_create": {
        "interface": "eth0",
        "ip_range": "192.168.1.0/24",
        "pool": "nova"
    }
}
```

1.36.3. Bulk-delete floating IPs

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-floating-ips-bulk/delete`	Bulk-deletes floating IPs.

Normal response codes: 200

1.36.3.1. Request

This table shows the URI parameters for the bulk-delete floating ips request:

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

Example 1.180. Bulk-delete floating IPs: JSON request

```
{
    "ip_range": "192.168.1.0/24"
}
```

1.36.3.2. Response

Example 1.181. Bulk-delete floating IPs: JSON response

```
{
    "floating_ips_bulk_delete": "192.168.1.0/24"
}
```

1.36.4. List floating IPs by host

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-floating-ips-bulk/{host_name}`	Lists all floating IPs for a host.

Normal response codes: 200

1.36.4.1. Request

This table shows the URI parameters for the list floating ips by host request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{host_name}`	String	The name of the host.

This operation does not accept a request body.

1.36.4.2. Response

Example 1.182. List floating IPs by host: JSON response

```
{
    "floating_ip_info": [
        {
            "address": "10.10.10.3",
            "instance_uuid": null,
            "fixed_ip": null,
            "interface": "eth0",
            "pool": "nova",
            "project_id": null
        }
    ]
}
```

1.37. Ping instances (os-fping)

Pings instances and reports which ones are alive.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-fping{?all_tenants,include,exclude}`	Run the fping utility to ping instances and report which ones are alive.
GET	`/v2.1/{tenant_id}/os-fping/{id}`	Run the fping utility to ping an instance and report whether it is alive.

1.37.1. Ping instances

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-fping{?all_tenants,include,exclude}`	Run the fping utility to ping instances and report which ones are alive.

Specify the `all_tenants=1` query parameter to ping instances for all tenants. For example:

```
GET /os-fping?all_tenants=1
```

Specify the `include` and `exclude` query parameters to filter the results. For example:

```
GET /os-fping?all_tenants=1&include=uuid1,uuid2&exclude=uuid3,uuid4
```

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.37.1.1. Request

This table shows the URI parameters for the ping instances request:

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

This table shows the query parameters for the ping instances request:

Name	Type	Description
`all_tenants`	Int *(Optional)*	Specify the `all_tenants=1` query parameter to ping instances for all tenants.
`include`	String *(Optional)*	Specify `include=uuid[,uuid...]` to include the instances in the results.
`exclude`	String *(Optional)*	Specify `exclude=uuid[,uuid...]` to exclude the instances from the results.

This operation does not accept a request body.

1.37.1.2. Response

Example 1.183. Ping instances: JSON response

```
{
    "servers": [
        {
            "alive": false,
            "id": "1d1aea35-472b-40cf-9337-8eb68480aaa1",
            "project_id": "openstack"
        }
    ]
}
```

1.37.2. Ping an instance

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-fping/{id}`	Run the fping utility to ping an instance and report whether it is alive.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.37.2.1. Request

This table shows the URI parameters for the ping an instance request:

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

This operation does not accept a request body.

1.37.2.2. Response

Example 1.184. Ping an instance: JSON response

```
{
    "server": {
        "alive": false,
        "id": "f5e6fd6d-c0a3-4f9e-aabf-d69196b6d11a",
        "project_id": "openstack"
    }
}
```

1.38. Hosts (os-hosts)

Manages physical hosts.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-hosts`	Lists hosts.
PUT	`/v2.1/{tenant_id}/os-hosts/{host_name}`	Enables or puts a host in maintenance mode.
GET	`/v2.1/{tenant_id}/os-hosts/{host_name}`	Shows details for a host.
GET	`/v2.1/{tenant_id}/os-hosts/{host_name}/reboot`	Reboots a host.
GET	`/v2.1/{tenant_id}/os-hosts/{host_name}/shutdown`	Shuts down a host.
GET	`/v2.1/{tenant_id}/os-hosts/{host_name}/startup`	Starts a host.

1.38.1. List hosts

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-hosts`	Lists hosts.

Normal response codes: 200

1.38.1.1. Request

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

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

This operation does not accept a request body.

1.38.1.2. Response

Example 1.185. List hosts: JSON response

```
{
    "hosts": [
        {
            "host_name": "b6e4adbc193d428ea923899d07fb001e",
            "service": "conductor",
            "zone": "internal"
        },
        {
            "host_name": "09c025b0efc64211bd23fc50fa974cdf",
            "service": "compute",
            "zone": "nova"
        },
        {
            "host_name": "a942ebfa00064d9d89a9e5a175cb9ba8",
            "service": "cert",
            "zone": "internal"
        },
        {
            "host_name": "e73ec0bd35c64de4a1adfa8b8969a1f6",
            "service": "consoleauth",
            "zone": "internal"
        },
        {
            "host_name": "396a8a0a234f476eb05fb9fbc5802ba7",
            "service": "network",
            "zone": "internal"
        },
        {
            "host_name": "abffda96592c4eacaf4111c28fddee17",
            "service": "scheduler",
            "zone": "internal"
        },
        {
            "host_name": "a8820f04962a4b4ba9fe2e9540c24094",
            "service": "cells",
            "zone": "internal"
        }
    ]
}
```

1.38.2. Enable host

Method	URI	Description
PUT	`/v2.1/{tenant_id}/os-hosts/{host_name}`	Enables or puts a host in maintenance mode.

Normal response codes: 200

1.38.2.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{host_name}`	String	The name of the host.

Example 1.186. Enable host: JSON request

```
{
    "status": "enable",
    "maintenance_mode": "disable"
}
```

1.38.2.2. Response

Example 1.187. Enable host: JSON response

```
{
    "host": "65c5d5b7e3bd44308e67fc50f362aee6",
    "maintenance_mode": "off_maintenance",
    "status": "enabled"
}
```

1.38.3. Show host details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-hosts/{host_name}`	Shows details for a host.

Normal response codes: 200

1.38.3.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{host_name}`	String	The name of the host.

This operation does not accept a request body.

1.38.3.2. Response

Example 1.188. 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)"
            }
        }
    ]
}
```

1.38.4. Reboot host

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-hosts/{host_name}/reboot`	Reboots a host.

Normal response codes: 200

1.38.4.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{host_name}`	String	The name of the host.

This operation does not accept a request body.

1.38.4.2. Response

Example 1.189. Reboot host: JSON response

```
{
    "host": "9557750dbc464741a89c907921c1cb31",
    "power_action": "reboot"
}
```

1.38.5. Shut down host

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-hosts/{host_name}/shutdown`	Shuts down a host.

Normal response codes: 200

1.38.5.1. Request

This table shows the URI parameters for the shut down host request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{host_name}`	String	The name of the host.

This operation does not accept a request body.

1.38.5.2. Response

Example 1.190. Shut down host: JSON response

```
{
    "host": "77cfa0002e4d45fe97f185968111b27b",
    "power_action": "shutdown"
}
```

1.38.6. Start host

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hosts/{host_name}/startup	Starts a host.

Normal response codes: 200

1.38.6.1. Request

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

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

This operation does not accept a request body.

1.38.6.2. Response

Example 1.191. Start host: JSON response

```
{
    "host": "4b392b27930343bbaa27fd5d8328a564",
    "power_action": "startup"
}
```

1.39. Hypervisors (os-hypervisors)

Lists all hypervisors, shows summary statistics for all hypervisors over all compute nodes, shows details for a hypervisor, and shows the uptime for a hypervisor.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hypervisors	Lists hypervisors.
GET	/v2.1/{tenant_id}/os-hypervisors/statistics	Shows summary statistics for all hypervisors over all compute nodes.
GET	/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}	Shows details for a hypervisor.
GET	/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}/uptime	Shows the uptime for a hypervisor.

1.39.1. List hypervisors

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-hypervisors`	Lists hypervisors.

Normal response codes: 200

1.39.1.1. Request

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

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

This operation does not accept a request body.

1.39.1.2. Response

Example 1.192. List hypervisors: JSON response

```
{
    "hypervisors": [
        {
            "status": "enabled",
            "state": "up",
            "id": 1,
            "hypervisor_hostname": "fake-mini"
        }
    ]
}
```

1.39.2. Show hypervisor statistics

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-hypervisors/statistics`	Shows summary statistics for all hypervisors over all compute nodes.

Normal response codes: 200

1.39.2.1. Request

This table shows the URI parameters for the show hypervisor statistics request:

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

This operation does not accept a request body.

1.39.2.2. Response

Example 1.193. Show hypervisor statistics: JSON response

```
{
    "hypervisor_statistics": {
        "count": 1,
        "vcpus_used": 0,
        "local_gb_used": 0,
        "memory_mb": 7980,
        "current_workload": 0,
        "vcpus": 8,
        "running_vms": 0,
        "free_disk_gb": 157,
        "disk_available_least": 140,
        "local_gb": 157,
        "free_ram_mb": 7468,
        "memory_mb_used": 512
    }
}
```

1.39.3. Show hypervisor details

Method	URI	Description
GET	/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}	Shows details for a hypervisor.

Normal response codes: 200

1.39.3.1. Request

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

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

This operation does not accept a request body.

1.39.3.2. Response

Example 1.194. Show hypervisor details: JSON response

```
{
    "hypervisor": {
        "status": "enabled",
        "service": {
            "host": "fake-mini",
            "disabled_reason": null,
            "id": 6
        },
        "vcpus_used": 0,
        "hypervisor_type": "QEMU",
        "local_gb_used": 0,
        "vcpus": 8,
        "hypervisor_hostname": "fake-mini",
        "memory_mb_used": 512,
        "memory_mb": 7980,
        "current_workload": 0,
        "state": "up",
        "host_ip": "23.253.248.171",
        "cpu_info": "{\"vendor\": \"Intel\", \"model\": \"gate64\", \"arch\":
 \"x86_64\", \"features\": [\"pge\", \"clflush\", \"sep\", \"syscall\", \"vme
\", \"msr\", \"cmov\", \"fpu\", \"pat\", \"lm\", \"tsc\", \"nx\", \"fxsr\",
 \"sse4.1\", \"pae\", \"sse4.2\", \"pclmuldq\", \"tsc-deadline\", \"mmx\", \
"cx8\", \"mce\", \"de\", \"rdtscp\", \"mca\", \"pse\", \"pni\", \"popcnt\", \
"apic\", \"sse\", \"lahf_lm\", \"aes\", \"sse2\", \"hypervisor\", \"ssse3\",
 \"cx16\", \"mtrr\", \"x2apic\"], \"topology\": {\"cores\": 1, \"cells\": 1, \
"threads\": 1, \"sockets\": 8}}",
        "running_vms": 0,
        "free_disk_gb": 157,
        "hypervisor_version": 2000000,
        "disk_available_least": 140,
        "local_gb": 157,
        "free_ram_mb": 7468,
        "id": 1
    }
}
```

1.39.4. Show hypervisor uptime

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-hypervisors/{hypervisor_id}/uptime`	Shows the uptime for a hypervisor.

Normal response codes: 200

1.39.4.1. Request

This table shows the URI parameters for the show hypervisor uptime request:

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

This operation does not accept a request body.

1.39.4.2. Response

Example 1.195. Show hypervisor uptime: JSON response

```
{
    "hypervisor": {
        "status": "enabled",
        "state": "up",
        "id": 1,
        "hypervisor_hostname": "fake-mini",
        "uptime": " 16:09:43 up 8 days, 19:58,  1 user,  load average: 0.86,
 0.63, 0.55\n"
    }
}
```

1.40. Instance usage audit log (os-instance-usage-audit-log)

Administrator only. Monitors task logs.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-instance_usage_audit_log`	Lists usage audits for an instance.
GET	`/v2.1/{tenant_id}/os-instance_usage_audit_log/{before_timestamp}{?before_timestamp}`	Lists usage audits that occurred before a specified time.

1.40.1. List usage audits for an instance

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-instance_usage_audit_log`	Lists usage audits for an instance.

Normal response codes: 200

1.40.1.1. Request

This table shows the URI parameters for the list usage audits for an instance request:

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

This operation does not accept a request body.

1.40.1.2. Response

Example 1.196. List usage audits for an instance: JSON response

```
{
    "instance_usage_audit_logs": {
        "hosts_not_run": [
            "f4eb7cfd155f4574967f8b55a7faed75"
        ],
        "log": {},
        "num_hosts": 1,
        "num_hosts_done": 0,
        "num_hosts_not_run": 1,
        "num_hosts_running": 0,
        "overall_status": "0 of 1 hosts done. 0 errors.",
        "period_beginning": "2012-12-01 00:00:00",
        "period_ending": "2013-01-01 00:00:00",
        "total_errors": 0,
        "total_instances": 0
    }
}
```

1.40.2. List usage audits before specified time

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-instance_usage_audit_log/{before_timestamp}{?before_timestamp}`	Lists usage audits that occurred before a specified time.

Normal response codes: 200

1.40.2.1. Request

This table shows the URI parameters for the list usage audits before specified time request:

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

This table shows the query parameters for the list usage audits before specified time request:

Name	Type	Description
`before_timestamp`	DateTime *(Required)*	The date and time before which to list usage audits. The date and time stamp format is ISO 8601: `CCYY-MM-DDThh:mm:ss±hh:mm` The `±hh:mm` value, if included, returns the time zone as an offset from UTC. For example, `2015-08-27T09:49:58-05:00`. If you omit the time zone, the UTC time zone is assumed.

This operation does not accept a request body.

1.40.2.2. Response

Example 1.197. List usage audits before specified time: JSON response

```
{
    "instance_usage_audit_log": {
        "hosts_not_run": [
            "8e33da2b48684ef3ab165444d6a7384c"
        ],
        "log": {},
        "num_hosts": 1,
        "num_hosts_done": 0,
        "num_hosts_not_run": 1,
        "num_hosts_running": 0,
        "overall_status": "0 of 1 hosts done. 0 errors.",
        "period_beginning": "2012-06-01 00:00:00",
        "period_ending": "2012-07-01 00:00:00",
        "total_errors": 0,
        "total_instances": 0
    }
}
```

1.41. Migrations (os-migrations)

Shows data on migrations.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-migrations`	Lists in-progress migrations.

1.41.1. List migrations

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-migrations`	Lists in-progress migrations.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.41.1.1. Request

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

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

This operation does not accept a request body.

1.41.1.2. Response

Example 1.198. List migrations: JSON response

```
{
    "migrations": [
        {
            "created_at": "2012-10-29T13:42:02.000000",
            "dest_compute": "compute2",
            "dest_host": "192.0.2.0",
            "dest_node": "node2",
            "id": 1234,
            "instance_uuid": "instance_id_123",
            "new_instance_type_id": 2,
            "old_instance_type_id": 1,
            "source_compute": "compute1",
            "source_node": "node1",
            "status": "Done",
            "updated_at": "2012-10-29T13:42:02.000000"
        },
        {
            "created_at": "2013-10-22T13:42:02.000000",
            "dest_compute": "compute20",
            "dest_host": "5.6.7.8",
            "dest_node": "node20",
            "id": 5678,
            "instance_uuid": "instance_id_456",
            "new_instance_type_id": 6,
            "old_instance_type_id": 5,
            "source_compute": "compute10",
            "source_node": "node10",
            "status": "Done",
            "updated_at": "2013-10-22T13:42:02.000000"
        }
    ]
}
```

1.42. Networks (os-networks)

Creates, lists, shows information for, and deletes networks.

Adds network to a project, disassociates a network from a project, and disassociates a project from a network.

Associates host with and disassociates host from a network.

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-networks`	Creates a network.
GET	`/v2.1/{tenant_id}/os-networks`	Lists networks for the project.
POST	`/v2.1/{tenant_id}/os-networks/add`	Adds a network to a project.
GET	`/v2.1/{tenant_id}/os-networks/{id}`	Shows details for a network.
DELETE	`/v2.1/{tenant_id}/os-networks/{id}`	Deletes a network.
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Associates a network with a host.
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Disassociates a host from a network.
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Disassociates a network from a project. You can then reuse the network.
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Disassociates a project from a network.

1.42.1. Create network

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-networks`	Creates a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.42.1.1. Request

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

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

Example 1.199. Create network: JSON request

```
{
    "network": {
        "label": "new net 111",
        "cidr": "10.20.105.0/24",
        "mtu": 9000,
        "dhcp_server": "10.20.105.2",
        "enable_dhcp": false,
        "share_address": true,
        "allowed_start": "10.20.105.10",
        "allowed_end": "10.20.105.200"
    }
}
```

1.42.1.2. Response

Example 1.200. Create network: JSON response

```
{
    "network": {
        "bridge": null,
        "bridge_interface": null,
        "broadcast": "10.20.105.255",
        "cidr": "10.20.105.0/24",
        "cidr_v6": null,
        "created_at": null,
        "deleted": null,
        "deleted_at": null,
        "dhcp_server": "10.20.105.2",
        "dhcp_start": "10.20.105.2",
        "dns1": null,
        "dns2": null,
        "enable_dhcp": false,
        "gateway": "10.20.105.1",
        "gateway_v6": null,
        "host": null,
        "id": "d7a17c0c-457e-4ab4-a99c-4fa1762f5359",
        "injected": null,
        "label": "new net 111",
        "mtu": 9000,
        "multi_host": null,
        "netmask": "255.255.255.0",
        "netmask_v6": null,
        "priority": null,
        "project_id": null,
        "rxtx_base": null,
        "share_address": true,
        "updated_at": null,
        "vlan": null,
        "vpn_private_address": null,
        "vpn_public_address": null,
        "vpn_public_port": null
    }
}
```

1.42.2. List networks

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-networks`	Lists networks for the project.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.42.2.1. Request

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

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

This operation does not accept a request body.

1.42.2.2. Response

Example 1.201. List networks: JSON response

```
{
    "networks": [
        {
            "bridge": "br100",
            "bridge_interface": "eth0",
            "broadcast": "10.0.0.7",
            "cidr": "10.0.0.0/29",
            "cidr_v6": null,
            "created_at": "2011-08-15T06:19:19.387525",
            "deleted": false,
            "deleted_at": null,
            "dhcp_server": "10.0.0.1",
            "dhcp_start": "10.0.0.3",
            "dns1": null,
            "dns2": null,
            "enable_dhcp": true,
            "gateway": "10.0.0.1",
            "gateway_v6": null,
            "host": "nsokolov-desktop",
            "id": "20c8acc0-f747-4d71-a389-46d078ebf047",
            "injected": false,
            "label": "mynet_0",
            "mtu": null,
            "multi_host": false,
            "netmask": "255.255.255.248",
            "netmask_v6": null,
            "priority": null,
            "project_id": "1234",
            "rxtx_base": null,
            "share_address": false,
            "updated_at": "2011-08-16T09:26:13.048257",
            "vlan": 100,
            "vpn_private_address": "10.0.0.2",
            "vpn_public_address": "127.0.0.1",
            "vpn_public_port": 1000
        },
        {
            "bridge": "br101",
            "bridge_interface": "eth0",
            "broadcast": "10.0.0.15",
            "cidr": "10.0.0.10/29",
            "cidr_v6": null,
            "created_at": "2011-08-15T06:19:19.885495",
            "deleted": false,
            "deleted_at": null,
            "dhcp_server": "10.0.0.9",
            "dhcp_start": "10.0.0.11",
            "dns1": null,
            "dns2": null,
            "enable_dhcp": true,
            "gateway": "10.0.0.9",
            "gateway_v6": null,
            "host": null,
            "id": "20c8acc0-f747-4d71-a389-46d078ebf000",
            "injected": false,
            "label": "mynet_1",
            "mtu": null,
            "multi_host": false,
            "netmask": "255.255.255.248",
            "netmask_v6": null,
            "priority": null,
            "project_id": null,
            "rxtx_base": null,
            "share_address": false,
            "updated_at": null,
            "vlan": 101,
            "vpn_private_address": "10.0.0.10",
            "vpn_public_address": null,
            "vpn_public_port": 1001
        }
    ]
}
```

1.42.3. Add network

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-networks/add`	Adds a network to a project.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.42.3.1. Request

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

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

Example 1.202. Add network: JSON request

```
{
    "id": "1"
}
```

1.42.4. Show network details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-networks/{id}`	Shows details for a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.42.4.1. Request

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

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

This operation does not accept a request body.

1.42.4.2. Response

Example 1.203. Show network details: JSON response

```
{
    "network": {
        "bridge": "br100",
        "bridge_interface": "eth0",
        "broadcast": "10.0.0.7",
        "cidr": "10.0.0.0/29",
        "cidr_v6": null,
        "created_at": "2011-08-15T06:19:19.387525",
        "deleted": false,
        "deleted_at": null,
        "dhcp_server": "10.0.0.1",
        "dhcp_start": "10.0.0.3",
        "dns1": null,
        "dns2": null,
        "enable_dhcp": true,
        "gateway": "10.0.0.1",
        "gateway_v6": null,
        "host": "nsokolov-desktop",
        "id": "20c8acc0-f747-4d71-a389-46d078ebf047",
        "injected": false,
        "label": "mynet_0",
        "mtu": null,
        "multi_host": false,
        "netmask": "255.255.255.248",
        "netmask_v6": null,
        "priority": null,
        "project_id": "1234",
        "rxtx_base": null,
        "share_address": false,
        "updated_at": "2011-08-16T09:26:13.048257",
        "vlan": 100,
        "vpn_private_address": "10.0.0.2",
        "vpn_public_address": "127.0.0.1",
        "vpn_public_port": 1000
    }
}
```

1.42.5. Delete network

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-networks/{id}`	Deletes a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.42.5.1. Request

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

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

This operation does not accept a request body.

1.42.6. Associate host

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Associates a network with a host.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.42.6.1. Request

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

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

Example 1.204. Associate host: JSON request

```
{
    "associate_host": "testHost"
}
```

1.42.7. Disassociate host

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Disassociates a host from a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.42.7.1. Request

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

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

Example 1.205. Disassociate host: JSON request

```
{
    "disassociate_host": null
}
```

1.42.8. Disassociate network

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Disassociates a network from a project. You can then reuse the network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.42.8.1. Request

This table shows the URI parameters for the disassociate network request:

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

Example 1.206. Disassociate network: JSON request

```
{
    "disassociate": null
}
```

1.42.9. Disassociate project

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-networks/{id}/action`	Disassociates a project from a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.42.9.1. Request

This table shows the URI parameters for the disassociate project request:

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

Example 1.207. Disassociate project: JSON request

```
{
    "disassociate_project": null
}
```

1.43. Quota class (os-quota-class-sets)

Provides quota classes management support.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-quota-class-sets/{class_id}`	Shows the quota for a class.
PUT	`/v2.1/{tenant_id}/os-quota-class-sets/{class_id}`	Updates quota for a class.

1.43.1. Show quota

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-quota-class-sets/{class_id}`	Shows the quota for a class.

Normal response codes: 200

1.43.1.1. Request

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

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

This operation does not accept a request body.

1.43.1.2. Response

Example 1.208. Show quota: JSON response

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

1.43.2. Update quota

Method	URI	Description
PUT	`/v2.1/{tenant_id}/os-quota-class-sets/{class_id}`	Updates quota for a class.

Normal response codes: 200

1.43.2.1. Request

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

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

Example 1.209. Update quota: JSON request

```
{
    "quota_class_set": {
        "instances": 50,
        "cores": 50,
        "ram": 51200,
        "floating_ips": 10,
        "metadata_items": 128,
        "injected_files": 5,
        "injected_file_content_bytes": 10240,
        "injected_file_path_bytes": 255,
        "security_groups": 10,
        "security_group_rules": 20,
        "key_pairs": 100
    }
}
```

1.43.2.2. Response

Example 1.210. Update quota: JSON response

```
{
    "quota_class_set": {
        "cores": 50,
        "fixed_ips": -1,
        "floating_ips": 10,
        "injected_file_content_bytes": 10240,
        "injected_file_path_bytes": 255,
        "injected_files": 5,
        "instances": 50,
        "key_pairs": 100,
        "metadata_items": 128,
        "ram": 51200,
        "security_group_rules": 20,
        "security_groups": 10
    }
}
```

1.44. Quota sets (os-quota-sets)

Permits administrators, depending on policy settings, to view quotas for a project and view and update default quotas.

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-quota-sets`	Deletes a quota for tenant.
PUT	`/v2.1/{tenant_id}/os-quota-sets`	Force-updates quota for tenant.
PUT	`/v2.1/{tenant_id}/os-quota-sets`	Updates quota for tenant.
GET	`/v2.1/{tenant_id}/os-quota-sets/defaults`	Shows default quotas for tenant.
GET	`/v2.1/{tenant_id}/os-quota-sets/detail`	Lists quotas with details for a tenant.
PUT	`/v2.1/{tenant_id}/os-quota-sets/{?user_id}`	Updates quota for user.
DELETE	`/v2.1/{tenant_id}/os-quota-sets/{?user_id}`	Deletes quota for a user.

1.44.1. Delete quota for tenant

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-quota-sets`	Deletes a quota for tenant.

Normal response codes: 202

1.44.1.1. Request

This table shows the URI parameters for the delete quota for tenant request:

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

This operation does not accept a request body.

1.44.2. Force-update quota

Method	URI	Description
PUT	`/v2.1/{tenant_id}/os-quota-sets`	Force-updates quota for tenant.

Normal response codes: 200

1.44.2.1. Request

This table shows the URI parameters for the force-update quota request:

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

Example 1.211. Force-update quota: JSON request

```
{
    "quota_set": {
        "force": "True",
        "instances": 45
    }
}
```

1.44.2.2. Response

Example 1.212. Force-update quota: JSON response

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

1.44.3. Update quota

Method	URI	Description
PUT	`/v2.1/{tenant_id}/os-quota-sets`	Updates quota for tenant.

Normal response codes: 200

1.44.3.1. Request

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

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

Example 1.213. Update quota: JSON request

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

1.44.3.2. Response

Example 1.214. Update quota: JSON response

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

1.44.4. Show default quotas

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-quota-sets/defaults`	Shows default quotas for tenant.

Normal response codes: 200

1.44.4.1. Request

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

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

This operation does not accept a request body.

1.44.4.2. Response

Example 1.215. Show default quotas: JSON response

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

1.44.5. Show quotas (detailed)

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-quota-sets/detail`	Lists quotas with details for a tenant.

Normal response codes: 200

1.44.5.1. Request

This table shows the URI parameters for the show quotas (detailed) request:

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

This operation does not accept a request body.

1.44.5.2. Response

Example 1.216. Show quotas (detailed): JSON response

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

1.44.6. Update quota for user

Method	URI	Description
PUT	/v2.1/{tenant_id}/os-quota-sets/{?user_id}	Updates quota for user.

Normal response codes: 200

1.44.6.1. Request

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

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

Example 1.217. Update quota for user: JSON request

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

1.44.6.2. Response

Example 1.218. Update quota for user: JSON response

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

1.44.7. Delete quota for user

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-quota-sets/{?user_id}`	Deletes quota for a user.

Normal response codes: 202

1.44.7.1. Request

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

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

This operation does not accept a request body.

1.45. Security groups (os-security-groups)

Lists, shows information for, creates, and deletes security groups.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-security-groups`	Lists security groups.
POST	`/v2.1/{tenant_id}/os-security-groups`	Creates a security group.
GET	`/v2.1/{tenant_id}/os-security-groups/{security_group_id}`	Shows details for a security group.
PUT	`/v2.1/{tenant_id}/os-security-groups/{security_group_id}`	Updates a security group.
DELETE	`/v2.1/{tenant_id}/os-security-groups/{security_group_id}`	Deletes a security group.
GET	`/v2.1/{tenant_id}/servers/{server_id}/os-security-groups`	Lists security groups for a server.

1.45.1. List security groups

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-security-groups`	Lists security groups.

Normal response codes: 200

1.45.1.1. Request

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

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

This operation does not accept a request body.

1.45.1.2. Response

Example 1.219. List security groups: JSON response

```
{
    "security_groups": [
        {
            "description": "default",
            "id": 1,
            "name": "default",
            "rules": [],
            "tenant_id": "openstack"
        }
    ]
}
```

1.45.2. Create security group

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-security-groups`	Creates a security group.

Normal response codes: 200

1.45.2.1. Request

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

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

Example 1.220. Create security group: JSON request

```
{
    "security_group": {
        "name": "test",
        "description": "description"
    }
}
```

1.45.2.2. Response

Example 1.221. Create security group: JSON response

```
{
    "security_group": {
        "description": "default",
        "id": 1,
        "name": "default",
        "rules": [],
        "tenant_id": "openstack"
    }
}
```

1.45.3. Show security group details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-security-groups/{security_group_id}`	Shows details for a security group.

Normal response codes: 200

1.45.3.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{security_group_id}`	Int	The ID of the security group.

This operation does not accept a request body.

1.45.3.2. Response

Example 1.222. Show security group: JSON response

```
{
    "security_group": {
        "description": "default",
        "id": 1,
        "name": "default",
        "rules": [],
        "tenant_id": "openstack"
    }
}
```

1.45.4. Update security group

Method	URI	Description
PUT	`/v2.1/{tenant_id}/os-security-groups/{security_group_id}`	Updates a security group.

Normal response codes: 200

1.45.4.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{security_group_id}`	Int	The ID of the security group.

Example 1.223. Update security group: JSON request

```
{
    "security_group": {
        "name": "mysecgroup",
        "description": "my security group"
    }
}
```

1.45.4.2. Response

Example 1.224. Update security group: JSON response

```
{
    "security_group": {
        "description": "my security group",
        "id": 1,
        "name": "mysecgroup",
        "rules": [],
        "tenant_id": "a52cdb9cc7854a39a23d3af73a40899e"
    }
}
```

1.45.5. Delete security group

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-security-groups/{security_group_id}`	Deletes a security group.

Normal response codes: 202

1.45.5.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{security_group_id}`	Int	The ID of the security group.

This operation does not accept a request body.

1.45.6. List security groups by server

Method	URI	Description
GET	/v2.1/{tenant_id}/servers/{server_id}/os-security-groups	Lists security groups for a server.

Normal response codes: 200

1.45.6.1. Request

This table shows the URI parameters for the list security groups by server request:

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

This operation does not accept a request body.

1.45.6.2. Response

Example 1.225. List security groups by server: JSON response

```
{
    "security_groups": [
        {
            "description": "default",
            "id": 1,
            "name": "default",
            "rules": [],
            "tenant_id": "openstack"
        }
    ]
}
```

1.46. Rules for default security group (os-security-group-default-rules)

Lists, shows information for, and creates default security group rules.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-security-group-default-rules	Lists default security group rules.
POST	/v2.1/{tenant_id}/os-security-group-default-rules	Creates a default security group rule.
GET	/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Shows details for a security group rule.
DELETE	/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}	Deletes a security group rule.

1.46.1. List default security group rules

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-security-group-default-rules`	Lists default security group rules.

Normal response codes: 200

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

1.46.1.1. Request

This table shows the URI parameters for the list default security group rules request:

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

This operation does not accept a request body.

1.46.1.2. Response

Example 1.226. 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
        }
    ]
}
```

1.46.2. Create default security group rule

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-security-group-default-rules`	Creates a default security group rule.

If you specify a source port (`from_port`) or destination port (`to_port`) value, you must specify an IP protocol (`ip_protocol`) value. Otherwise, the operation returns the Bad `Request (400)` response code.

Normal response codes: 200

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

1.46.2.1. Request

This table shows the URI parameters for the create default security group rule request:

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

Example 1.227. Create default security group rule: JSON request

```
{
    "security_group_default_rule": {
        "ip_protocol": "TCP",
        "from_port": "80",
        "to_port": "80",
        "cidr": "10.10.10.0/24"
    }
}
```

1.46.2.2. Response

Example 1.228. 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
    }
}
```

1.46.3. Show default security group rule details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}`	Shows details for a security group rule.

Normal response codes: 200

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

1.46.3.1. Request

This table shows the URI parameters for the show default security group rule details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{security_group_default_rule_id}`	UUID	The UUID of the security group rule.

This operation does not accept a request body.

1.46.3.2. Response

Example 1.229. Show default security group rule: JSON response

```
{
    "security_group_default_rule": {
        "from_port": 80,
        "id": 1,
        "ip_protocol": "TCP",
        "ip_range": {
            "cidr": "10.10.10.0/24"
        },
        "to_port": 80
    }
}
```

1.46.4. Delete default security group rule

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-security-group-default-rules/{security_group_default_rule_id}`	Deletes a security group rule.

Normal response codes: 204

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

1.46.4.1. Request

This table shows the URI parameters for the delete default security group rule request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{security_group_default_rule_id}`	UUID	The UUID of the security group rule.

This operation does not accept a request body.

1.47. Rules for security group (os-security-group-rules)

Creates and deletes security group rules.

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-security-group-rules`	Creates a rule for a security group.
DELETE	`/v2.1/{tenant_id}/os-security-group-rules/{security_group_rule_id}`	Deletes a security group rule.

1.47.1. Create security group rule

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-security-group-rules`	Creates a rule for a security group.

Normal response codes: 200

1.47.1.1. Request

This table shows the URI parameters for the create security group rule request:

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

Example 1.230. Create security group rule: JSON request

```
{
    "security_group_rule": {
        "from_port": "443",
        "ip_protocol": "tcp",
        "to_port": "443",
        "cidr": "0.0.0.0/0",
        "parent_group_id": "48700ff3-30b8-4e63-845f-a79c9633e9fb"
    }
}
```

1.47.1.2. Response

Example 1.231. Create security group rule: JSON response

```
{
    "security_group_rule": {
        "id": "1",
        "ip_range": {
            "cidr": "0.0.0.0/0"
        },
        "parent_group_id": "48700ff3-30b8-4e63-845f-a79c9633e9fb",
        "to_port": 443,
        "ip_protocol": "tcp",
        "group": {},
        "from_port": 443
    }
}
```

1.47.2. Delete security group rule

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-security-group-rules/{security_group_rule_id}`	Deletes a security group rule.

Normal response codes: 202

1.47.2.1. Request

This table shows the URI parameters for the delete security group rule request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{security_group_rule_id}`	Int	The ID of the security group rule.

This operation does not accept a request body.

1.48. Create external events (os-server-external-events)

Creates one or more external events. The API dispatches each event to a server instance.

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-server-external-events`	Creates one or more external events, which the API dispatches to the instance.

1.48.1. Run events

Method	URI	Description
POST	/v2.1/{tenant_id}/os-server-external-events	Creates one or more external events, which the API dispatches to the instance.

You must assign this instance to a host. Otherwise, this call does not dispatch the event to the instance.

Normal response codes: 200

1.48.1.1. Request

This table shows the URI parameters for the run events request:

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

Example 1.232. Run events: JSON request

```
{
    "events": [
        {
            "name": "test-event",
            "tag": "foo",
            "status": "completed",
            "server_uuid": "3df201cf-2451-44f2-8d25-a4ca826fc1f3"
        }
    ]
}
```

1.48.1.2. Response

Example 1.233. Run events: JSON response

```
{
    "events": [
        {
            "code": 200,
            "name": "network-changed",
            "server_uuid": "ff1df7b2-6772-45fd-9326-c0a3b05591c2",
            "status": "completed",
            "tag": "foo"
        }
    ]
}
```

1.49. Server groups (os-server-groups)

Lists, shows information for, creates, and deletes server groups.

Method	URI	Description
GET	/v2.1/{tenant_id}/os-server-groups{?all_projects}	Lists all server groups for the tenant.

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-server-groups`	Creates a server group.
GET	`/v2.1/{tenant_id}/os-server-groups/{server_group_id}`	Shows details for a server group.
DELETE	`/v2.1/{tenant_id}/os-server-groups/{server_group_id}`	Deletes a server group.

1.49.1. List server groups

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-server-groups{?all_projects}`	Lists all server groups for the tenant.

Administrative users can use the `all_projects` query parameter to list all server groups for all projects.

Normal response codes: 200

1.49.1.1. Request

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

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

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

Name	Type	Description
`all_projects`	Boolean *(Optional)*	Administrator only. Lists server groups for all projects. For example: `GET /v2.1/{admin_tenant_id}/os-server-groups?all_projects=True` If you specify a tenant ID for a non-administrative user with this query parameter, the call lists all server groups for the tenant, or project, rather than for all projects.

This operation does not accept a request body.

1.49.1.2. Response

Example 1.234. List server groups: JSON response

```
{
    "server_groups": [
        {
            "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
            "name": "test",
            "policies": [
                "anti-affinity"
            ],
            "members": [],
            "metadata": {}
        }
    ]
}
```

1.49.2. Create server group

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-server-groups`	Creates a server group.

Normal response codes: 200

1.49.2.1. Request

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

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

Example 1.235. Create server group: JSON request

```
{
    "server_group": {
        "name": "test",
        "policies": [
            "anti-affinity"
        ]
    }
}
```

1.49.2.2. Response

Example 1.236. Create server group: JSON response

```
{
    "server_group": {
        "id": "5bbcc3c4-1da2-4437-a48a-66f15b1b13f9",
        "name": "test",
        "policies": [
            "anti-affinity"
        ],
        "members": [],
        "metadata": {}
    }
}
```

1.49.3. Show server group details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-server-groups/{server_group_id}`	Shows details for a server group.

Normal response codes: 200

1.49.3.1. Request

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

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

This operation does not accept a request body.

1.49.3.2. Response

Example 1.237. Show server group details: JSON response

```
{
    "server_group": {
        "id": "5bbcc3c4-1da2-4437-a48a-66f15b1b13f9",
        "name": "test",
        "policies": [
            "anti-affinity"
        ],
        "members": [],
        "metadata": {}
    }
}
```

1.49.4. Delete server group

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-server-groups/{server_group_id}`	Deletes a server group.

Normal response codes: 204

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

1.49.4.1. Request

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

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

This operation does not accept a request body.

1.50. Usage reports (os-simple-tenant-usage)

Reports usage statistics on compute and storage resources.

Method	URI	Description
GET	`/v2.1/os-simple-tenant-usage`	Lists usage information for all tenants.
GET	`/v2.1/os-simple-tenant-usage/{tenant_id}`	Shows usage details for a tenant.

1.50.1. List tenant usage for all tenants

Method	URI	Description
GET	`/v2.1/os-simple-tenant-usage`	Lists usage information for all tenants.

Normal response codes: 200

1.50.1.1. Response

Example 1.238. List tenant usage for all tenants: JSON response

```
{
    "tenant_usages": [
        {
            "start": "2012-10-08T21:10:44.587336",
            "stop": "2012-10-08T22:10:44.587336",
            "tenant_id": "openstack",
            "total_hours": 1.0,
            "total_local_gb_usage": 1.0,
            "total_memory_mb_usage": 512.0,
            "total_vcpus_usage": 1.0
        }
    ]
}
```

1.50.2. Show usage details for tenant

Method	URI	Description
GET	/v2.1/os-simple-tenant-usage/{tenant_id}	Shows usage details for a tenant.

Normal response codes: 200

1.50.2.1. Request

This table shows the URI parameters for the show usage details for tenant request:

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

1.50.2.2. Response

Example 1.239. Show usage details for tenant: JSON response

```
{
    "tenant_usage": {
        "server_usages": [
            {
                "ended_at": null,
                "flavor": "m1.tiny",
                "hours": 1.0,
                "instance_id": "1f1deceb-17b5-4c04-84c7-e0d4499c8fe0",
                "local_gb": 1,
                "memory_mb": 512,
                "name": "new-server-test",
                "started_at": "2012-10-08T20:10:44.541277",
                "state": "active",
                "tenant_id": "openstack",
                "uptime": 3600,
                "vcpus": 1
            }
        ],
        "start": "2012-10-08T20:10:44.587336",
        "stop": "2012-10-08T21:10:44.587336",
        "tenant_id": "openstack",
        "total_hours": 1.0,
        "total_local_gb_usage": 1.0,
        "total_memory_mb_usage": 512.0,
        "total_vcpus_usage": 1.0
    }
}
```

1.51. Project networks (os-tenant-networks)

Creates, lists, shows information for, and deletes project networks.

Method	URI	Description
POST	/v2.1/{tenant_id}/os-tenant-networks	Creates a project network.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-tenant-networks`	Lists all project networks.
GET	`/v2.1/{tenant_id}/os-tenant-networks/{id}`	Shows details for a project network.
DELETE	`/v2.1/{tenant_id}/os-tenant-networks/{id}`	Deletes a project network.
POST	`/v2.1/{tenant_id}/os-tenant-networks/{id}/action`	Associates a network with a host.
POST	`/v2.1/{tenant_id}/os-tenant-networks/{id}/action`	Disassociates a host from a network.
POST	`/v2.1/{tenant_id}/os-tenant-networks/{id}/action`	Disassociates a network from a project so that the network can be reused.
POST	`/v2.1/{tenant_id}/os-tenant-networks/{id}/action`	Disassociates a project from a network.

1.51.1. Create project network

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-tenant-networks`	Creates a project network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.51.1.1. Request

This table shows the URI parameters for the create project network request:

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

Example 1.240. Create project network: JSON request

```
{
    "network": {
        "label": "public",
        "cidr": "172.0.0.0/24",
        "vlan_start": 1,
        "num_networks": 1,
        "network_size": 255
    }
}
```

1.51.1.2. Response

Example 1.241. Create project network: JSON response

```
{
    "network": {
        "cidr": "172.0.0.0/24",
        "id": "5bbcc3c4-1da2-4437-a48a-66f15b1b13f9",
        "label": "public"
    }
}
```

1.51.2. List project networks

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-tenant-networks`	Lists all project networks.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.51.2.1. Request

This table shows the URI parameters for the list project networks request:

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

This operation does not accept a request body.

1.51.2.2. Response

Example 1.242. List project networks: JSON response

```
{
    "networks": [
        {
            "bridge": "br100",
            "bridge_interface": "eth0",
            "broadcast": "10.0.0.7",
            "cidr": "10.0.0.0/29",
            "cidr_v6": null,
            "created_at": "2011-08-15T06:19:19.387525",
            "deleted": false,
            "deleted_at": null,
            "dhcp_server": "10.0.0.1",
            "dhcp_start": "10.0.0.3",
            "dns1": null,
            "dns2": null,
            "enable_dhcp": true,
            "gateway": "10.0.0.1",
            "gateway_v6": null,
            "host": "nsokolov-desktop",
            "id": "20c8acc0-f747-4d71-a389-46d078ebf047",
            "injected": false,
            "label": "mynet_0",
            "mtu": null,
            "multi_host": false,
            "netmask": "255.255.255.248",
            "netmask_v6": null,
            "priority": null,
            "project_id": "1234",
            "rxtx_base": null,
            "share_address": false,
            "updated_at": "2011-08-16T09:26:13.048257",
            "vlan": 100,
            "vpn_private_address": "10.0.0.2",
            "vpn_public_address": "127.0.0.1",
            "vpn_public_port": 1000
        },
        {
            "bridge": "br101",
            "bridge_interface": "eth0",
            "broadcast": "10.0.0.15",
            "cidr": "10.0.0.10/29",
            "cidr_v6": null,
            "created_at": "2011-08-15T06:19:19.885495",
            "deleted": false,
            "deleted_at": null,
            "dhcp_server": "10.0.0.9",
            "dhcp_start": "10.0.0.11",
            "dns1": null,
            "dns2": null,
            "enable_dhcp": true,
            "gateway": "10.0.0.9",
            "gateway_v6": null,
            "host": null,
            "id": "20c8acc0-f747-4d71-a389-46d078ebf000",
            "injected": false,
            "label": "mynet_1",
            "mtu": null,
            "multi_host": false,
            "netmask": "255.255.255.248",
            "netmask_v6": null,
            "priority": null,
            "project_id": null,
            "rxtx_base": null,
            "share_address": false,
            "updated_at": null,
            "vlan": 101,
            "vpn_private_address": "10.0.0.10",
            "vpn_public_address": null,
            "vpn_public_port": 1001
        }
    ]
}
```

1.51.3. Show project network details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-tenant-networks/{id}`	Shows details for a project network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 200

1.51.3.1. Request

This table shows the URI parameters for the show project network details request:

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

This operation does not accept a request body.

1.51.3.2. Response

Example 1.243. Show project network details: JSON response

```
{
    "networks": [
        {
            "cidr": "10.0.0.0/29",
            "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
            "label": "test_0"
        },
        {
            "cidr": "10.0.0.8/29",
            "id": "616fb98f-46ca-475e-917e-2563e5a8cd20",
            "label": "test_1"
        }
    ]
}
```

1.51.4. Delete project network

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-tenant-networks/{id}`	Deletes a project network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.51.4.1. Request

This table shows the URI parameters for the delete project network request:

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

This operation does not accept a request body.

1.51.5. Associate host

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-tenant-networks/{id}/action`	Associates a network with a host.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.51.5.1. Request

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

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

Example 1.244. Associate host: JSON request

```
{
    "associate_host": "testHost"
}
```

1.51.6. Disassociate host

Method	URI	Description
POST	/v2.1/{tenant_id}/os-tenant-networks/{id}/action	Disassociates a host from a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the policy.json file.

Normal response codes: 202

1.51.6.1. Request

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

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

Example 1.245. Disassociate host: JSON request

```
{
    "disassociate_host": null
}
```

1.51.7. Disassociate network

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-tenant-networks/{id}/action`	Disassociates a network from a project so that the network can be reused.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.51.7.1. Request

This table shows the URI parameters for the disassociate network request:

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

Example 1.246. Disassociate network: JSON request

```
{
    "disassociate": null
}
```

1.51.8. Disassociate project

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-tenant-networks/{id}/action`	Disassociates a project from a network.

Policy defaults enable only users with the administrative role or the owner of the server to perform this operation. Cloud providers can change these permissions through the `policy.json` file.

Normal response codes: 202

1.51.8.1. Request

This table shows the URI parameters for the disassociate project request:

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

Example 1.247. Disassociate project: JSON request

```
{
    "disassociate_project": null
}
```

1.52. Volume extension (os-volumes, os-snapshots)

Manages volumes and snapshots for use with the Compute API.

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-volumes`	Lists the volumes associated with the account.
POST	`/v2.1/{tenant_id}/os-volumes`	Creates a volume.
GET	`/v2.1/{tenant_id}/os-volumes/detail`	Lists all volumes with details.
GET	`/v2.1/{tenant_id}/os-volumes/{volume_id}`	Shows details for a volume.
DELETE	`/v2.1/{tenant_id}/os-volumes/{volume_id}`	Deletes a volume.
GET	`/v2.1/{tenant_id}/os-volume-types`	Lists volume types.
GET	`/v2.1/{tenant_id}/os-volume-types/{volume_type_id}`	Shows details for a volume type.
POST	`/v2.1/{tenant_id}/os-snapshots`	Creates a snapshot.
GET	`/v2.1/{tenant_id}/os-snapshots`	Lists snapshots.
GET	`/v2.1/{tenant_id}/os-snapshots/detail`	Lists all snapshots with details.
GET	`/v2.1/{tenant_id}/os-snapshots/{snapshot_id}`	Shows details for a snapshot.

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-snapshots/{snapshot_id}`	Deletes a snapshot from the account.

1.52.1. List volumes

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-volumes`	Lists the volumes associated with the account.

Normal response codes: 200

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

1.52.1.1. Request

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

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

This operation does not accept a request body.

1.52.1.2. Response

Example 1.248. 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"
        }
    ]
}
```

1.52.2. Create volume

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-volumes`	Creates a volume.

Normal response codes: 200

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

1.52.2.1. Request

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

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

Example 1.249. 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"
    }
}
```

1.52.2.2. Response

Example 1.250. Create volume: JSON response

```
{
    "volume": {
        "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
        "display_name": "vol-001",
        "display_description": "Another volume.",
        "status": "active",
        "size": 30,
        "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
        "metadata": {
            "contents": "junk"
        },
        "availability_zone": "us-east1",
        "snapshot_id": null,
        "attachments": [],
        "created_at": "2012-02-14T20:53:07Z"
    }
}
```

1.52.3. List volumes with details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-volumes/detail`	Lists all volumes with details.

Normal response codes: 200

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

1.52.3.1. Request

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

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

This operation does not accept a request body.

1.52.3.2. Response

Example 1.251. List volumes with details: JSON response

```
{
    "volumes": [
        {
            "attachments": [
                {
                    "device": "/",
                    "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
                    "serverId": "3912f2b4-c5ba-4aec-9165-872876fe202e",
                    "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
                }
            ],
            "availabilityZone": "zone1:host1",
            "createdAt": "1999-01-01T01:01:01.000000",
            "displayDescription": "Volume Description",
            "displayName": "Volume Name",
            "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
            "metadata": {},
            "size": 100,
            "snapshotId": null,
            "status": "in-use",
            "volumeType": "Backup"
        }
    ]
}
```

1.52.4. Show volume details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-volumes/{volume_id}`	Shows details for a volume.

Normal response codes: 200

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

1.52.4.1. Request

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

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

This operation does not accept a request body.

1.52.4.2. Response

Example 1.252. Show volume details: JSON response

```
{
    "volume": {
        "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
        "display_name": "vol-001",
        "display_description": "Another volume.",
        "status": "active",
        "size": 30,
        "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
        "metadata": {
            "contents": "junk"
        },
        "availability_zone": "us-east1",
        "snapshot_id": null,
        "attachments": [],
        "created_at": "2012-02-14T20:53:07Z"
    }
}
```

1.52.5. Delete volume

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-volumes/{volume_id}`	Deletes a volume.

Normal response codes: 202

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

1.52.5.1. Request

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

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

This operation does not accept a request body.

1.52.5.2. Response

Example 1.253. Delete volume: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Date: Fri, 05 Dec 2014 00:39:32 GMT
```

This operation does not return a response body.

1.52.6. List volume types

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-volume-types`	Lists volume types.

Normal response codes: 200

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

1.52.6.1. Request

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

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

This operation does not accept a request body.

1.52.6.2. Response

Example 1.254. 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": {}
        }
    ]
}
```

1.52.7. Show volume type details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-volume-types/{volume_type_id}`	Shows details for a volume type.

Normal response codes: 200

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

1.52.7.1. Request

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

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_type_id}`	String	The unique ID for a volume type.

This operation does not accept a request body.

1.52.7.2. Response

Example 1.255. Show volume type details: JSON response

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

1.52.8. Create snapshot

Method	URI	Description
POST	`/v2.1/{tenant_id}/os-snapshots`	Creates a snapshot.

Normal response codes: 200

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

1.52.8.1. Request

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

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

Example 1.256. Create snapshot: JSON request

```
{
    "snapshot": {
        "createdAt": "2013-02-25T16:27:54.680544",
        "displayDescription": "Daily backup",
        "displayName": "snap-001",
        "id": 100,
        "size": 100,
        "status": "available",
        "volumeId": "521752a6-acf6-4b2d-bc7a-119f9148cd8c"
    }
}
```

1.52.8.2. Response

Example 1.257. Create snapshot: JSON response

```
{
    "snapshot": {
        "createdAt": "2013-02-25T16:27:54.724209",
        "displayDescription": "Default description",
        "displayName": "Default name",
        "id": "100",
        "size": 100,
        "status": "available",
        "volumeId": 12
    }
}
```

1.52.9. List snapshots

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-snapshots`	Lists snapshots.

Normal response codes: 200

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

1.52.9.1. Request

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

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

This operation does not accept a request body.

1.52.9.2. Response

Example 1.258. List snapshots: JSON response

```
{
    "snapshots": [
        {
            "createdAt": "2013-02-25T16:27:54.684999",
            "displayDescription": "Default description",
            "displayName": "Default name",
            "id": 100,
            "size": 100,
            "status": "available",
            "volumeId": 12
        },
        {
            "createdAt": "2013-02-25T16:27:54.685005",
            "displayDescription": "Default description",
            "displayName": "Default name",
            "id": 101,
            "size": 100,
            "status": "available",
            "volumeId": 12
        },
        {
            "createdAt": "2013-02-25T16:27:54.685008",
            "displayDescription": "Default description",
            "displayName": "Default name",
            "id": 102,
            "size": 100,
            "status": "available",
            "volumeId": 12
        }
    ]
}
```

1.52.10. List snapshots with details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-snapshots/detail`	Lists all snapshots with details.

Normal response codes: 200

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

1.52.10.1. Request

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

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

This operation does not accept a request body.

1.52.10.2. Response

Example 1.259. List snapshots with details: JSON response

```
{
    "snapshots": [
        {
            "createdAt": "2013-02-25T16:27:54.684999",
            "displayDescription": "Default description",
            "displayName": "Default name",
            "id": 100,
            "size": 100,
            "status": "available",
            "volumeId": 12
        },
        {
            "createdAt": "2013-02-25T16:27:54.685005",
            "displayDescription": "Default description",
            "displayName": "Default name",
            "id": 101,
            "size": 100,
            "status": "available",
            "volumeId": 12
        },
        {
            "createdAt": "2013-02-25T16:27:54.685008",
            "displayDescription": "Default description",
            "displayName": "Default name",
            "id": 102,
            "size": 100,
            "status": "available",
            "volumeId": 12
        }
    ]
}
```

1.52.11. Show snapshot details

Method	URI	Description
GET	`/v2.1/{tenant_id}/os-snapshots/{snapshot_id}`	Shows details for a snapshot.

Normal response codes: 200

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

1.52.11.1. Request

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

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

This operation does not accept a request body.

1.52.11.2. Response

Example 1.260. Show snapshot details: JSON response

```
{
    "snapshot": {
        "createdAt": "2013-02-25T16:27:54.724209",
        "displayDescription": "Default description",
        "displayName": "Default name",
        "id": "100",
        "size": 100,
        "status": "available",
        "volumeId": 12
    }
}
```

1.52.12. Delete snapshot

Method	URI	Description
DELETE	`/v2.1/{tenant_id}/os-snapshots/{snapshot_id}`	Deletes a snapshot from the account.

This operation is asynchronous. You must list snapshots repeatedly to determine whether the snapshot was deleted.

Normal response codes: 202

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

1.52.12.1. Request

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

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

This operation does not accept a request body.

1.52.12.2. Response

Example 1.261. Delete snapshot: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Date: Mon, 01 Dec 2014 16:23:10 GMT
```

This operation does not return a response body.