

OpenStack Compute

API v2 and Extensions Reference

API v2 (April 21, 2014)



OpenStack Compute API v2 and Extensions Reference

API v2 (2014-04-21)

Copyright © 2009-2014 OpenStack Foundation All rights reserved.

This document is intended for software developers interested in developing applications using the OpenStack Compute Application Programming Interface (API).

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

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

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

Table of Contents

Preface	18
1. Intended audience	18
2. Document change history	19
3. Additional resources	21
1. General API information	1
1.1. Concepts	1
1.2. Authentication	2
1.3. Request and response types	2
1.4. Links and references	6
1.5. Paginated collections	12
1.6. Efficient polling with the <i>Changes-Since</i> parameter	17
1.7. Limits	18
1.8. Versions	23
1.9. Extensions	32
1.10. Faults	58
2. API operations	65
2.1. Servers	72
2.2. Server addresses	94
2.3. Server actions	98
2.4. Flavors	109
2.5. Images	114
2.6. Metadata	124
2.7. Networks	142
3. Compute API v2 extensions	143
3.1. Server admin actions (action)	152
3.2. Server console output (os-console-output)	169
3.3. Server console (os-consoles)	170
3.4. Server deferred delete (os-deferred-delete)	173
3.5. Server diagnostics (diagnostics)	175
3.6. Flavor access (flavors)	176
3.7. Flavors with FlavorDisabled attribute (flavors)	189
3.8. Flavor extra-specs (os-extra-specs)	194
3.9. Flavors with rxtx_factor extended attribute (flavors)	200
3.10. Flavors with extended attributes (flavors)	207
3.11. Flavors create or delete (flavors)	215
3.12. Images with size attribute (images)	219
3.13. Limits with project usage (limits)	228
3.14. Limits with project usage for administrators (limits)	231
3.15. Guest agents (os-agents)	235
3.16. Host aggregates (os-aggregates)	242
3.17. Attach interfaces (os-attach-interfaces)	257
3.18. Root certificates (os-certificates)	263
3.19. Cloudpipe (os-cloudpipe)	268
3.20. Coverage reports (os-coverage)	271
3.21. Fixed IPs (os-fixed-ips)	275
3.22. Floating IP DNS records (os-floating-ip-dns)	278
3.23. Floating IP pools (os-floating-ip-pools)	287
3.24. Floating IPs (os-floating-ips)	288

3.25. Floating IPs bulk (os-floating-ips-bulk)	296
3.26. Hosts (os-hosts)	302
3.27. Hypervisors (os-hypervisors)	311
3.28. Server actions (os-instance-actions)	318
3.29. Keypairs (os-keypairs)	323
3.30. Migrations (os-migrations)	328
3.31. Networks (os-networks)	330
3.32. Quota sets (os-quota-sets)	343
3.33. Server rescue and unrescue (os-rescue)	355
3.34. Rules for default security group (os-security-group-default-rules)	357
3.35. Security groups (os-security-groups)	361
3.36. Server password (os-server-password)	368
3.37. Server shelve (os-server-shelve)	371
3.38. Server start and stop (os-server-start-stop)	372
3.39. Manage services (os-services)	374
3.40. Usage reports (os-simple-tenant-usage)	381
3.41. Virtual interfaces (os-virtual-interfaces)	384
3.42. Volume extension (os-volumes, os-snapshots)	386
3.43. Volume attachments (os-volume_attachments)	405
3.44. Servers with block device mapping format (servers)	411
3.45. Server OS-EXT-IPS-MAC:mac_addr extended attribute (servers)	416
3.46. Configuration drive (servers)	425
3.47. Servers with extended availability zones (servers)	434
3.48. Servers and images with disk config (servers, images)	439
3.49. Server IP type (servers)	462
3.50. Server extended attributes (servers)	467
3.51. Server extended status (servers)	473
3.52. Servers multiple create (servers)	479
3.53. Servers with scheduler hints (servers)	483

List of Tables

1.1. Sample rate limits	18
1.2. Sample Absolute Limits	19
1.3. Fault Elements and Error Codes	59
2.1. Network attributes	142

List of Examples

1.1. JSON request with headers	3
1.2. XML response with headers	4
1.3. JSON request with XML query extension for the response	4
1.4. ID image reference: JSON request	7
1.5. ID image reference: XML request	7
1.6. Full image reference: JSON request	7
1.7. Full image reference: XML request	8
1.8. Server with self links: JSON	10
1.9. Server with self links: XML	10
1.10. Server with alternate link: JSON	11
1.11. Image with alternate link: XML	11
1.12. Images collection: XML (first page)	13
1.13. Images collection: JSON (first page)	13
1.14. Images collection: XML (second page)	14
1.15. Images collection: JSON (second page)	14
1.16. Images collection: XML (last page)	15
1.17. Images collection: JSON (last page)	15
1.18. Paginated image metadata: XML	16
1.19. Paginated image metadata: JSON	16
1.20. List limits: JSON response	21
1.21. List limits: XML response	22
1.22. Request with MIME type versioning	23
1.23. Request with URI versioning	23
1.24. Multiple choices: XML response	25
1.25. Multiple choices: JSON response	26
1.26. List versions: request	27
1.27. List versions: XML response	27
1.28. List versions: Atom response	28
1.29. List versions: JSON response	28
1.30. Get version details: request	29
1.31. Get version details: XML response	29
1.32. Get version details: Atom response	30
1.33. Get version details: JSON response	31
1.34. Extended server: XML response	32
1.35. Extended server: JSON response	33
1.36. Extended action: XML response	35
1.37. Extended action: JSON response	35
1.38. List extensions: JSON response	36
1.39. List extensions: XML response	48
1.40. Get extension: JSON response	56
1.41. Get extension: XML response	56
1.42. Fault: XML response	58
1.43. Fault: JSON response	58
1.44. Item Not Found fault: JSON response	59
1.45. Item Not Found fault: XML response	59
1.46. Over Limit fault: XML response	60
1.47. Over Limit fault: JSON response	60
1.48. Server in error state: XML response	61

1.49. Server in error state: JSON response	62
1.50. Image in error state: XML response	63
1.51. Image in error state: JSON response	64
2.1. List servers: JSON response	74
2.2. List servers: XML response	75
2.3. Create server: JSON request	77
2.4. Create server: XML request	77
2.5. Create server: JSON response	78
2.6. Create server: XML response	78
2.7. Create server with access IP: XML request	80
2.8. Create server with access IP: JSON request	81
2.9. Create server with multiple access IPs: XML request	81
2.10. Create server with multiple access IPs: JSON request	82
2.11. Get server details: JSON response	83
2.12. Get server details: XML response	84
2.13. Update Server Name Request: JSON	87
2.14. Update Server Name Request: XML	87
2.15. Update Server IP Addresses Request: JSON	87
2.16. Update Server IP Addresses Request: XML	87
2.17. Update Server Name Response: JSON	88
2.18. Update Server Name Response: XML	89
2.19. Update Server IP Addresses Response: JSON	90
2.20. Update Server IP Addresses Response: XML	91
2.21. List addresses: JSON response	95
2.22. List addresses: XML response	96
2.23. List addresses by network: JSON response	97
2.24. List addresses by network: XML response	98
2.25. Change password: JSON request	99
2.26. Change password: XML request	99
2.27. Reboot server: JSON request	100
2.28. Reboot server: XML request	100
2.29. Rebuild server: JSON request	101
2.30. Rebuild server: XML request	101
2.31. Rebuild server: JSON response	102
2.32. Rebuild server: XML response	103
2.33. Resize server: JSON request	105
2.34. Resize server: XML request	105
2.35. Confirm resized server: JSON request	106
2.36. Confirm resized server: XML request	106
2.37. Revert resized server: JSON request	107
2.38. Revert resized server: XML request	107
2.39. Create image: JSON request	108
2.40. Create image: XML request	108
2.41. List details for flavors: JSON response	110
2.42. List details for flavors: XML response	112
2.43. Get flavor details: JSON response	113
2.44. Get flavor details: XML response	113
2.45. List images details: JSON response	115
2.46. List images details: XML response	119
2.47. Get image details: JSON response	122
2.48. Get image details: XML response	123

2.49. Show server metadata: JSON response	125
2.50. Show server metadata: XML response	125
2.51. Create or replace server metadata: JSON request	126
2.52. Create or replace server metadata: XML request	126
2.53. Create or replace server metadata: JSON response	126
2.54. Create or replace server metadata: XML response	126
2.55. Update server metadata items: JSON request	128
2.56. Update server metadata items: XML request	128
2.57. Update server metadata items: JSON response	128
2.58. Update server metadata items: XML response	128
2.59. Show server metadata item details: JSON response	130
2.60. Show server metadata item details: XML response	130
2.61. Create or update server metadata item: JSON request	131
2.62. Create or update server metadata item: XML request	131
2.63. Create or update server metadata item: JSON response	131
2.64. Create or update server metadata item: XML response	131
2.65. Show image metadata: JSON response	134
2.66. Show image metadata: XML response	134
2.67. Create or replace image metadata: JSON request	135
2.68. Create or replace image metadata: XML request	135
2.69. Create or replace image metadata: JSON response	135
2.70. Create or replace image metadata: XML response	136
2.71. Update image metadata items: JSON request	137
2.72. Update image metadata items: XML request	137
2.73. Update image metadata items: JSON response	137
2.74. Update image metadata items: XML response	138
2.75. Show image metadata item details: JSON response	139
2.76. Show image metadata item details: XML response	139
2.77. Create or update image metadata item: JSON request	140
2.78. Create or update image metadata item: XML request	140
2.79. Create or update image metadata item: JSON response	140
2.80. Create or update image metadata item: XML response	141
3.1. Pause server: JSON request	154
3.2. Pause server: XML request	154
3.3. Unpause server: JSON request	155
3.4. Unpause server: XML request	155
3.5. Suspend server: JSON request	156
3.6. Suspend server: XML request	156
3.7. Resume server: JSON request	157
3.8. Resume server: XML request	157
3.9. Migrate server: JSON request	158
3.10. Migrate server: xml request	158
3.11. Reset network: JSON request	159
3.12. Reset network: XML request	159
3.13. Insert network information: JSON request	160
3.14. Insert network information: XML request	160
3.15. Lock server: JSON request	161
3.16. Lock server: XML request	161
3.17. Unlock server: JSON request	162
3.18. Unlock server: XML request	162
3.19. Create server backup: JSON request	163

3.20. Create server backup: XML request	163
3.21. Live-migrate server: JSON request	164
3.22. Live-migrate server: XML request	164
3.23. Reset server state: JSON request	165
3.24. Reset server state: XML request	165
3.25. Evacuate server: JSON request	166
3.26. Evacuate server: XML request	166
3.27. Evacuate server: JSON response	166
3.28. Evacuate server: XML response	166
3.29. Add security group: JSON request	167
3.30. Add security group: XML request	167
3.31. Remove security group: JSON request	168
3.32. Remove security group: XML request	168
3.33. Add floating IP address: JSON request	169
3.34. Get console output: JSON request	170
3.35. Get console output: XML request	170
3.36. Get console output: JSON response	170
3.37. Get console output: XML response	170
3.38. Get console: JSON request	172
3.39. Get console: XML request	172
3.40. Get console: JSON response	172
3.41. Get console: XML response	172
3.42. Force delete server: JSON request	174
3.43. Force delete server: XML request	174
3.44. Restore server: JSON request	175
3.45. Restore server: XML request	175
3.46. Server diagnostics: JSON response	176
3.47. List flavors with access type: JSON response	178
3.48. List flavors with access type: XML response	180
3.49. Create private flavor: JSON request	181
3.50. Create private flavor: XML request	181
3.51. Create private flavor: JSON response	181
3.52. Create private flavor: XML response	182
3.53. Show flavor access type: JSON response	183
3.54. Show flavor access type: XML response	183
3.55. List tenants with access to private flavor: JSON response	185
3.56. List tenants with access to private flavor: XML response	185
3.57. Add access to private flavor: JSON request	186
3.58. Add access to private flavor: XML request	186
3.59. Add access to private flavor: JSON response	186
3.60. Add access to private flavor: XML response	186
3.61. Delete access from private flavor: JSON request	188
3.62. Delete access from private flavor: XML request	188
3.63. Delete access from private flavor: JSON response	188
3.64. Delete access from private flavor: XML response	188
3.65. Get flavor disabled status details: JSON response	190
3.66. Get flavor disabled status details: XML response	190
3.67. List flavors with flavor disabled status: JSON response	192
3.68. List flavors with flavor disabled status: XML response	194
3.69. List flavor extra specs: JSON response	196
3.70. List flavor extra specs: XML response	196

3.71. Create flavor extra specs: JSON request	197
3.72. Create flavor extra specs: XML request	197
3.73. Create flavor extra specs: JSON response	197
3.74. Create flavor extra specs: XML response	197
3.75. Get flavor extra spec details: JSON response	199
3.76. Get flavor extra spec details: XML response	199
3.77. Create flavor with rxtx_factor: JSON request	201
3.78. Create flavor with rxtx_factor: XML request	201
3.79. Create flavor with rxtx_factor: JSON response	201
3.80. Create flavor with rxtx_factor: XML response	202
3.81. Get flavor with rxtx_factor: JSON response	203
3.82. Get flavor with rxtx_factor: XML response	203
3.83. Get flavor Details with rxtx_factor: JSON response	205
3.84. Get flavor Details with rxtx_factor: XML response	207
3.85. Create flavor with extra data: JSON request	209
3.86. Create flavor with extra data: XML request	209
3.87. Create flavor with extra data: JSON response	209
3.88. Create flavor with extra data: XML response	210
3.89. Get flavor extra data details: JSON response	211
3.90. Get flavor extra data details: XML response	211
3.91. List flavors with extra data: JSON response	213
3.92. List flavors with extra data: XML response	215
3.93. Create flavor: JSON request	217
3.94. Create flavor: XML request	217
3.95. Create flavor: JSON response	217
3.96. Create flavor: XML response	218
3.97. List details for images: JSON response	220
3.98. List details for images: XML response	224
3.99. Get image details: JSON response	227
3.100. Get image details: XML response	228
3.101. Get limits: JSON response	229
3.102. Get limits: XML response	230
3.103. Used limits for admins: JSON response	233
3.104. Used limits for admins: XML response	235
3.105. List agents: JSON response	237
3.106. List agents: XML response	237
3.107. Create agent: JSON request	238
3.108. Create agent: XML request	238
3.109. Create agent: JSON response	238
3.110. Create agent: XML response	239
3.111. Update agent: JSON request	241
3.112. Update agent: XML request	241
3.113. Update agent: JSON response	241
3.114. Update agent: XML response	241
3.115. List aggregates: XML response	243
3.116. List aggregates: JSON response	243
3.117. Create aggregate: XML request	245
3.118. Create aggregate: JSON request	245
3.119. Create aggregate: XML response	245
3.120. Create aggregate: JSON response	245
3.121. Get aggregate details: XML response	248

3.122. Get aggregate details: JSON response	248
3.123. Update aggregate: XML request	250
3.124. Update aggregate: JSON request	250
3.125. Update aggregate: XML response	250
3.126. Update aggregate: JSON response	250
3.127. Set aggregate metadata: XML request	252
3.128. Set aggregate metadata: JSON request	252
3.129. Set aggregate metadata: XML response	252
3.130. Set aggregate metadata: JSON response	253
3.131. Add host to aggregate: XML request	254
3.132. Add host to aggregate: JSON request	254
3.133. Add host to aggregate: XML response	254
3.134. Add host to aggregate: JSON response	254
3.135. Remove host from aggregate: XML request	256
3.136. Remove host from aggregate: JSON request	256
3.137. Remove host from aggregate: XML response	256
3.138. Remove host from aggregate: JSON response	256
3.139. Create interface: JSON request	258
3.140. Create interface: XML request	258
3.141. Create interface: JSON response	258
3.142. Create interface: XML response	258
3.143. List interfaces: JSON response	260
3.144. List interfaces: XML response	260
3.145. Show attached interface information: JSON response	262
3.146. Show attached interface information: XML response	262
3.147. Create root certificate: JSON response	264
3.148. Create root certificate: XML response	265
3.149. Show root certificate details: JSON response	267
3.150. Show root certificate details: XML response	267
3.151. List cloudpipes: JSON response	269
3.152. List cloudpipes: XML response	269
3.153. Create cloudpipe: JSON request	270
3.154. Create cloudpipe: XML request	270
3.155. Create cloudpipe: JSON response	270
3.156. Create cloudpipe: XML response	270
3.157. Update cloudpipe: JSON request	271
3.158. Update cloudpipe: XML request	271
3.159. Start combined report: JSON request	272
3.160. Start combined report: XML request	272
3.161. Get coverage report: JSON response	272
3.162. Get coverage report: XML response	272
3.163. Start coverage report: JSON request	273
3.164. Start coverage report: XML request	273
3.165. Start combined report: JSON request	274
3.166. Start combined report: XML request	274
3.167. Stop coverage report: JSON request	275
3.168. Stop coverage report: XML request	275
3.169. Stop report: JSON response	275
3.170. Stop report: XML response	275
3.171. Show fixed IP information: JSON response	277
3.172. Show fixed IP information: XML response	277

3.173. Reserve or release a fixed IP: JSON request	278
3.174. Reserve or release a fixed IP: XML request	278
3.175. List DNS domains: JSON response	279
3.176. List DNS domains: XML response	279
3.177. Create or update DNS domain: JSON request	280
3.178. Create or update DNS domain: XML request	280
3.179. Create or update DNS domain: JSON response	280
3.180. Create or update DNS domain: XML response	280
3.181. Create or update DNS entry: JSON request	283
3.182. Create or update DNS entry: XML request	283
3.183. Create or update DNS entry: JSON response	283
3.184. Create or update DNS entry: XML response	283
3.185. Find unique DNS entry: JSON response	285
3.186. Find unique DNS entry: XML response	285
3.187. List DNS entries: JSON response	287
3.188. List DNS entries: XML response	287
3.189. List floating IP pools: JSON response	288
3.190. List floating IP pools: XML response	288
3.191. List floating IPs: JSON response	290
3.192. List floating IPs: XML response	290
3.193. Allocate floating IP: JSON request	291
3.194. Allocate floating IP: XML request	291
3.195. Allocate floating IP: JSON response	291
3.196. Allocate floating IP: XML response	291
3.197. Show floating IP information: JSON response	293
3.198. Show floating IP information: XML response	293
3.199. Add floating IP: JSON request	295
3.200. Add floating IP: XML request	295
3.201. Remove floating IP: JSON request	296
3.202. Remove floating IP: XML request	296
3.203. List floating IPs: JSON response	297
3.204. List floating IPs: XML response	297
3.205. Create floating IPs: JSON request	299
3.206. Create floating IPs: XML request	299
3.207. Create floating IPs: JSON response	299
3.208. Create floating IPs: XML response	299
3.209. Bulk-delete floating IPs: JSON request	301
3.210. Bulk-delete floating IPs: XML request	301
3.211. Bulk-delete floating IPs: JSON response	301
3.212. Bulk-delete floating IPs: XML response	301
3.213. List floating IPs by host: JSON response	302
3.214. List floating IPs by host: XML response	302
3.215. List hosts: JSON response	304
3.216. List hosts: XML response	305
3.217. Show host information: JSON response	306
3.218. Show host information: XML response	307
3.219. Update host: JSON request	308
3.220. Update host: XML request	308
3.221. Update host: JSON response	308
3.222. Update host: XML response	308
3.223. Start host: JSON response	309

3.224. Start host: XML response	309
3.225. Shut down host: JSON response	310
3.226. Shut down host: XML response	310
3.227. Reboot host: JSON response	311
3.228. Reboot host: XML response	311
3.229. List hypervisors: JSON response	313
3.230. List hypervisors: XML response	313
3.231. Show hypervisor information: JSON response	314
3.232. Show hypervisor information: XML response	314
3.233. Show statistics for hypervisors: JSON response	316
3.234. Show statistics for hypervisors: XML response	316
3.235. Show hypervisor up time: JSON response	317
3.236. Show hypervisor up time: XML response	317
3.237. List instances for hypervisors: JSON response	318
3.238. List instances for hypervisors: XML response	318
3.239. List server actions: JSON response	320
3.240. List server actions: XML response	320
3.241. Get action details: JSON response	322
3.242. Get action details: XML response	323
3.243. List keypairs: JSON response	324
3.244. List keypairs: XML response	324
3.245. Create or import keypair: XML request	325
3.246. Create or import keypair: JSON request	325
3.247. Create or import keypair: XML response	325
3.248. Create or import keypair: JSON response	325
3.249. Show keypair information: JSON response	328
3.250. Get migrations: JSON response	329
3.251. Get migrations: XML response	330
3.252. Create network: JSON request	331
3.253. Create network: XML request	331
3.254. Create network: JSON response	331
3.255. Create network: XML response	332
3.256. List networks: JSON response	333
3.257. List networks: XML response	334
3.258. Add network: JSON request	336
3.259. Add network: XML request	336
3.260. Show network information: JSON response	337
3.261. Show network information: XML response	338
3.262. Associate host: JSON request	340
3.263. Associate host: XML request	340
3.264. Disassociate host: JSON request	341
3.265. Disassociate host: XML request	341
3.266. Disassociate network: JSON request	342
3.267. Disassociate network: XML request	342
3.268. Disassociate project: JSON request	343
3.269. Disassociate project: XML request	343
3.270. Show quotas response: JSON	344
3.271. Show quotas response: XML	344
3.272. Update quotas response: JSON	346
3.273. Show quotas response: XML	346
3.274. Update quota response: JSON	346

3.275. Update quota response: XML	347
3.276. Get default quotas response: JSON	348
3.277. Get default quotas response: XML	348
3.278. Show quotas for user response: JSON	350
3.279. Show quotas for user response: XML	350
3.280. Update quotas for user request: JSON	352
3.281. Update quotas for user request: XML	352
3.282. Update quotas for user response: JSON	352
3.283. Show quotas for user response: XML	353
3.284. Show quota details for user response: JSON	354
3.285. Rescue server: JSON request	356
3.286. Rescue server: XML request	356
3.287. Rescue server: JSON response	356
3.288. Rescue server: XML response	356
3.289. Unrescue server: JSON request	357
3.290. Unrescue server: XML request	357
3.291. List default security group rules: JSON response	358
3.292. List default security group rules: XML response	358
3.293. Create default security group rule: JSON request	359
3.294. Create default security group rule: XML request	359
3.295. Create default security group rule: JSON response	359
3.296. Create default security group rule: XML response	360
3.297. Show default security group rule: JSON response	361
3.298. Show default security group rule: XML response	361
3.299. List security groups: JSON response	363
3.300. List security group: XML response	363
3.301. Create security group: JSON request	364
3.302. Create security group: XML request	364
3.303. Create security group: JSON response	364
3.304. Create security group: XML response	364
3.305. List security groups by server: JSON response	366
3.306. List security groups by server: XML response	366
3.307. Show security group: JSON response	367
3.308. Show security group: XML response	367
3.309. Get server password: JSON response	369
3.310. Get server password: XML response	369
3.311. Clear server password: JSON request	370
3.312. Clear server password: XML request	370
3.313. Clear server password: JSON response	371
3.314. Clear server password: XML response	371
3.315. Shelve server: JSON request	372
3.316. Shelve server: XML request	372
3.317. Start server: JSON request	373
3.318. Start server: XML request	373
3.319. Stop server: JSON request	374
3.320. Stop server: XML request	374
3.321. List services: JSON response	375
3.322. Enable scheduling for a service: JSON request	377
3.323. Enable scheduling for a service: JSON response	377
3.324. Disable scheduling for a service: JSON request	378
3.325. Disable scheduling for a service: JSON response	378

3.326. Log disabled service information: JSON request	379
3.327. Log disabled service information: XML request	379
3.328. Log disabled service information: JSON response	379
3.329. Log disabled service information: XML response	379
3.330. List disabled services: JSON response	380
3.331. List disabled services: XML response	381
3.332. List usage information for all tenants: JSON response	382
3.333. List usage information for all tenants: XML response	382
3.334. Get tenant usage information: JSON response	383
3.335. Get tenant usage information: XML response	383
3.336. List virtual interfaces: JSON response	385
3.337. List virtual interfaces: XML response	385
3.338. Show virtual interface and attached network: JSON response	386
3.339. Show virtual interface and attached network: XML response	386
3.340. List volumes: JSON response	388
3.341. List volumes: XML response	388
3.342. List details for volumes: JSON response	390
3.343. List details for volumes: XML response	390
3.344. Create volume: JSON request	392
3.345. Create volume: XML request	392
3.346. Create volume: JSON response	392
3.347. Create volume: XML response	393
3.348. Show volume information: JSON response	394
3.349. Show volume information: XML response	394
3.350. Delete volume: JSON response	395
3.351. List volume types: JSON response	396
3.352. List volume types: XML response	396
3.353. Show volume type: JSON response	397
3.354. Show volume type: XML response	397
3.355. Create snapshot: JSON request	398
3.356. Create snapshot: XML request	398
3.357. Create snapshot: JSON response	398
3.358. Create snapshot: XML response	398
3.359. List snapshots: JSON response	400
3.360. List snapshots: XML response	400
3.361. List details for snapshots: JSON response	402
3.362. List details for snapshots: XML response	402
3.363. Show snapshot: JSON response	404
3.364. Show snapshot: XML response	404
3.365. Delete snapshot: JSON response	405
3.366. Attach volume: XML request	406
3.367. Attach volume: JSON request	406
3.368. Attach volume: XML response	406
3.369. Attach volume: JSON response	406
3.370. List volume attachments: XML response	408
3.371. List volume attachments: JSON response	408
3.372. Show volume attachment details: XML response	410
3.373. Show volume attachment details: JSON response	410
3.374. List servers: JSON response	412
3.375. List servers: XML response	413
3.376. Create server: JSON request	414

3.377. Create server: XML request	415
3.378. Create server: JSON response	415
3.379. Create server: XML response	416
3.380. Create server with OS-EXT-IPS-MAC:mac_addr extended attribute: JSON request	417
3.381. Create server with OS-EXT-IPS-MAC:mac_addr extended attribute: XML request	418
3.382. Create server with OS-EXT-IPS-MAC:mac_addr extended attribute: JSON response	418
3.383. Create server with OS-EXT-IPS-MAC:mac_addr extended attribute: XML response	418
3.384. Show server information: JSON response	420
3.385. Show server information: XML response	421
3.386. Get server details: JSON response	423
3.387. Get server details: XML response	424
3.388. Create server with configuration drive: JSON request	426
3.389. Create server with configuration drive: XML request	426
3.390. Create server with configuration drive: JSON response	427
3.391. Create server with configuration drive: XML response	427
3.392. Get server information with configuration drive: JSON response	429
3.393. Get server information with configuration drive: XML response	430
3.394. Get server details with configuration drive: JSON response	432
3.395. Get server details with configuration drive: XML response	433
3.396. Show server: JSON response	435
3.397. Show server: XML response	436
3.398. List details for servers: JSON response	437
3.399. List details for servers: XML response	438
3.400. Create server: JSON request	440
3.401. Create server: XML request	440
3.402. Create server: JSON response	441
3.403. Create server: XML response	441
3.404. Show server information: JSON response	442
3.405. Show server information: XML response	443
3.406. Update server: JSON request	444
3.407. Update server: XML request	444
3.408. Update server: JSON response	444
3.409. Update server: XML response	445
3.410. Resize server: JSON request	447
3.411. Resize server: XML request	447
3.412. Rebuild server: JSON request	448
3.413. Rebuild server: XML request	448
3.414. Rebuild server: JSON response	448
3.415. Rebuild server: XML response	449
3.416. List servers: JSON response	451
3.417. List servers: XML response	452
3.418. Get image information: JSON response	454
3.419. Get image information: XML response	455
3.420. List images: JSON response	456
3.421. List images: XML response	460
3.422. Show IP type: JSON response	463
3.423. Show IP type: XML response	464

3.424. List servers with IP type: JSON response	465
3.425. List servers with IP type: XML response	466
3.426. List servers with extended server attributes: JSON response	468
3.427. List servers with extended server attributes: XML response	469
3.428. Show extended server attributes: JSON response	471
3.429. Show extended server attributes: XML response	472
3.430. Show server extended status: JSON response	474
3.431. Show server extended status: XML response	475
3.432. List extended status for servers: JSON response	477
3.433. List extended status for servers: XML response	478
3.434. Create multiple servers: JSON request	480
3.435. Create multiple servers: XML request	481
3.436. Create multiple servers: JSON request	481
3.437. Create multiple servers: XML request	482
3.438. Create multiple servers: JSON response	482
3.439. Create multiple servers: XML response	482
3.440. Create multiple servers: JSON response	483
3.441. Create multiple servers: XML response	483
3.442. Create server with scheduler hints: JSON request	484
3.443. Create server with scheduler hints: XML request	484
3.444. Create server with scheduler hints: JSON response	485
3.445. Create server with scheduler hints: XML response	485

Preface

1. Intended audience	18
2. Document change history	19
3. Additional resources	21

OpenStack Compute is a compute service that provides server capacity in the cloud. Compute Servers come in different flavors of memory, cores, disk space, and CPU, and can be provisioned in minutes. Interactions with Compute Servers can occur programmatically via the OpenStack Compute API.

We welcome feedback, comments, and bug reports at bugs.launchpad.net/nova.

1. Intended audience

This guide assists software developers who want to develop applications using the OpenStack Compute API. To use this information, you should have access to an account from an OpenStack Compute provider, and you should also be familiar with the following concepts:

- OpenStack Compute service
- ReSTful web services
- HTTP/1.1
- JSON and/or XML data serialization formats

2. Document change history

This version of the Developer Guide replaces and obsoletes all previous versions. The most recent changes are described in the table below:

Revision Date	Summary of Changes
September 10, 2013	<ul style="list-style-type: none"> Added the 422 return code to the reboot server operation.
May 22, 2013	<ul style="list-style-type: none"> Updated the book title for consistency.
April 27, 2013	<ul style="list-style-type: none"> Changed title of this book to reflect that it contains Compute extensions.
April 17, 2013	<ul style="list-style-type: none"> Added the server admin actions extension.
May 30, 2012	<ul style="list-style-type: none"> Added more descriptive information about the request body attributes to the create server API operation.
May 2, 2012	<ul style="list-style-type: none"> Added SHUTOFF to the list of server status values for the list servers API operation. Updated the description for the SUSPENDED server status value for the list servers API operation.
April 24, 2012	<ul style="list-style-type: none"> Updated to use color-coded syntax formatting in request and response examples.
March 25, 2012	<ul style="list-style-type: none"> Added descriptions of URI parameters and request body attributes for API operations.
February 14, 2012	<ul style="list-style-type: none"> Updated the API version from v1.1 to v2. No longer use mimetype parameters to denote version.
November 8, 2011	<ul style="list-style-type: none"> Removed DRAFT designation.
September 8, 2011	<ul style="list-style-type: none"> Added <i>limit</i> and <i>marker</i> parameters to list operations. The rebuild action behaves just like create: an imageRef is used and a password may be specified. Added tenant and user_id attributes to server and image. Added vcpus attribute to flavors. The flavorRef attribute is now used in the resize action.
July 23, 2011	<ul style="list-style-type: none"> Added missing response examples for server update. Ensure consistent HTTP status codes for all resources. Clarifications on setting and changing a server password. Minor updates to metadata section for clarity. Discuss alternate links. Removed version number from compute media types — use a media type parameter instead. Bought back the flavorRef and imageRef server attributes these are now only used when creating a server. Made the create image operation a server action. Added minDisk and minRam filters to flavor lists. Added minDisk and minRam attributes to images. Asynchronous faults may now contain a timestamp. Changes-since request returns an empty list rather than a 304. Added DELETED image status. Fix content length in Example 1.2, "XML response with headers" [4]. Fixed bad request error code in Section 2.1.2.2, "Server passwords" [79]. Compact image, server, and flavor lists should contain IDs, names, and links (Any kind of link may be included — not just self links). Changed metadata URI from <code>.../meta</code> to <code>.../metadata</code> for consistency.
June 29, 2011	<ul style="list-style-type: none"> Renamed Primary IP to Access IP.
June 23, 2011	<ul style="list-style-type: none"> Many minor updates based on community feedback. Removed sections on Content Compression, Persistent Connections, and Caching — these are operator specific. Added section on HTTP. A Location header is returned when creating servers/images. Added filters to collection of Image, Servers, and Flavors. Added asynchronous faults. Updates to links and references. Remove serverRef, imageRef, and flavorRef and instead embed one entity in another to provide links. Added primary IP addresses. Added forbidden fault. We now use a single bookmark link per entity regardless of mimetype.

Revision Date	Summary of Changes
	<ul style="list-style-type: none"> • Collections are now sorted by create time. • Previous links are no longer required. • Added the ability to create or update multiple metadata items simultaneously. • Minor cleanups to server and image state machine. • Update to JSON collection format. • Replace integer IDs with UUIDs. • Removed affinityID, this will likely come in as an extension.
April 25, 2011	<ul style="list-style-type: none"> • Some minor cleanups in preparation for OpenStack Summit discussion.
March 11, 2011	<ul style="list-style-type: none"> • Many minor updates based on community feedback. • Updates to resource linking and references. • Better description of paginated collections. • Metadata supported in servers and images. • Dropped support for shared IP groups. • IPs organized by network ID, versus showing only public and private IPs. • Generalized affinity ID.
February 9, 2011	<ul style="list-style-type: none"> • Initial release.

3. Additional resources

You can download the most current version of this document from the OpenStack Docs website at <http://docs.openstack.org>.

1. General API information

1.1. Concepts	1
1.2. Authentication	2
1.3. Request and response types	2
1.4. Links and references	6
1.5. Paginated collections	12
1.6. Efficient polling with the <i>Changes-Since</i> parameter	17
1.7. Limits	18
1.8. Versions	23
1.9. Extensions	32
1.10. Faults	58

The OpenStack Compute API is defined as a ReSTful HTTP service. The API takes advantage of all aspects of the HTTP protocol (methods, URIs, media types, response codes, etc.) and providers are free to use existing features of the protocol such as caching, persistent connections, and content compression among others. For example, providers who employ a caching layer may respond with a 203 when a request is served from the cache instead of a 200. Additionally, providers may offer support for conditional **GET** requests using ETags, or they may send a redirect in response to a **GET** request. Clients should be written to account for these differences.

Providers can return information identifying requests in HTTP response headers, for example, to facilitate communication between the provider and client users.

1.1. Concepts

To use the OpenStack Compute API effectively, you should understand several key concepts:

- **Server**

A virtual machine (VM) instance in the compute system. Flavor and image are requisite elements when creating a server.

- **Flavor**

An available hardware configuration for a server. Each flavor has a unique combination of disk space, memory capacity and priority for CPU time.

- **Image**

A collection of files used to create or rebuild a server. Operators provide a number of pre-built OS images by default. You may also create custom images from cloud servers you have launched. These custom images are useful for backup purposes or for producing “gold” server images if you plan to deploy a particular server configuration frequently.

- **Reboot**

Use this function to perform either a soft or hard reboot of a server. With a soft reboot, the operating system is signaled to restart, which allows for a graceful shutdown of all

processes. A hard reboot is the equivalent of power cycling the server. The virtualization platform should ensure that the reboot action has completed successfully even in cases in which the underlying domain/VM is paused or halted/stopped.

- **Rebuild**

Use this function to remove all data on the server and replaces it with the specified image. Server ID and IP addresses remain the same.

- **Resize**

Use this function to convert an existing server to a different flavor, in essence, scaling the server up or down. The original server is saved for a period of time to allow rollback if there is a problem. All resizes should be tested and explicitly confirmed, at which time the original server is removed. All resizes are automatically confirmed after 24 hours if you do not confirm or revert them.

1.2. Authentication

Each HTTP request against the OpenStack Compute system requires the inclusion of specific authentication credentials. A single deployment may support multiple authentication schemes (OAuth, Basic Auth, Token). The authentication scheme used is determined by the provider of the OpenStack Compute system. Please contact your provider to determine the best way to authenticate against this API.



Note

Some authentication schemes may require that the API operate using SSL over HTTP (HTTPS).

1.3. Request and response types

The OpenStack Compute API supports both JSON and XML data serialization request and response formats.

You specify the request format in the `Content-Type` header in the request. This header is required for operations that have a request body. The syntax for the `Content-Type` header is:

```
Content-Type: application/format
```

Where *format* is either `json` or `xml`, you specify the response format by using one of the following methods:

- `Accept` header. The syntax for the `Accept` header is:

```
Accept: application/format
```

Where *format* is either `json` or `xml`, the default format is `json`.

- `Query extension`. Add an `.xml` or `.json` extension to the request URI. For example, the `.xml` extension in the following URI request specifies that the response body is returned in XML format:

[Example 1.2, "XML response with headers" \[4\]](#) shows the headers and XML response returned by the JSON request:

Example 1.2. XML response with headers

```
HTTP/1.1 202 Accepted
Date: Mon, 23 Jul 2012 20:24:48 GMT
Content-Length: 582
Location: https://servers.api.openstack.org/v2/010101/servers/06dba123-2c7e-4639-bea0-09fbe219b056
Content-Type: application/xml
X-Compute-Request-Id: req-ab05045a-452f-4b46-be0d-86494da02a2b
Server: Jetty(8.0.y.z-SNAPSHOT)
```

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  id="ea8417a1-7957-4ed5-8b3c-0befc1724308" adminPass="FoFw84XtQj3c">
  <metadata/>
  <atom:link
    href="http://openstack.example.com/v2/openstack/servers/
ea8417a1-7957-4ed5-8b3c-0befc1724308"
    rel="self"/>
  <atom:link
    href="http://openstack.example.com/openstack/servers/ea8417a1-7957-4ed5-8b3c-0befc1724308"
    rel="bookmark"/>
</server>
```

The following example shows an alternative method of achieving the same result. The following request uses an URI extension of `.xml` instead of an `Accept` header to request an XML response.



Note

The XML response is not shown.

Example 1.3. JSON request with XML query extension for the response

```
POST /v2/010101/servers.xml HTTP/1.1
Host: servers.api.openstack.org
Content-Type: application/json
X-Auth-Token: eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

```
{
  "server": {
    "flavorRef": "http://openstack.example.com/openstack/flavors/1",
    "imageRef": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-
b260-868f441e862b",
    "metadata": {
      "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "personality": [
      {
        "path": "/etc/banner.txt"
      }
    ],
    "networks": [
      {
        "uuid": "c6a48469-4638-4346-b755-6e66ed7abdfb",
```

```
    "fixed_ip": "10.0.0.10"  
  }  
} }  
}
```

1.4. Links and references

Often resources need to refer to other resources. For example, when creating a server, you must specify the image from which to build the server. You can specify the image by providing an ID or a URL to a remote image. When providing an ID, it is assumed that the resource exists in the current OpenStack deployment.

For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources:

- A `self` link contains a versioned link to the resource. Use these links when the link will be followed immediately.
- A `bookmark` link provides a permanent link to a resource that is appropriate for long term storage.
- An `alternate` link can contain an alternate representation of the resource. For example, an OpenStack Compute image might have an alternate representation in the OpenStack Image service.



Note

The `type` attribute provides a hint as to the type of representation to expect when following the link.

Example 1.8. Server with self links: JSON

```
{
  "server" : {
    "id" : "52415800-8b69-11e0-9b19-734fcede0043",
    "name" : "my-server",
    "links" : [
      {
        "rel" : "self",
        "href" : "http://servers.api.openstack.org/v2/1234/
servers/52415800-8b69-11e0-9b19-734fcede0043"
      },
      {
        "rel" : "bookmark",
        "href" : "http://servers.api.openstack.org/1234/servers/
52415800-8b69-11e0-9b19-734fcede0043"
      }
    ]
  }
}
```

Example 1.9. Server with self links: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  id="52415800-8b69-11e0-9b19-734fcede0043" name="my-server">
  <atom:link
    rel="self"
    href="http://servers.api.openstack.org/v2/1234/servers/
52415800-8b69-11e0-9b19-734fcede0043"/>
  <atom:link
    rel="bookmark"
    href="http://servers.api.openstack.org/1234/servers/
52415800-8b69-11e0-9b19-734fcede0043"/>
</server>
```

Example 1.10. Server with alternate link: JSON

```
{
  "image" : {
    "id" : "52415800-8b69-11e0-9b19-734f5736d2a2",
    "name" : "My Server Backup",
    "links": [
      {
        "rel" : "self",
        "href" : "http://servers.api.openstack.org/v2/1234/images/52415800-8b69-11e0-9b19-734f5736d2a2"
      },
      {
        "rel" : "bookmark",
        "href" : "http://servers.api.openstack.org/1234/images/52415800-8b69-11e0-9b19-734f5736d2a2"
      },
      {
        "rel" : "alternate",
        "type" : "application/vnd.openstack.image",
        "href" : "http://glance.api.openstack.org/1234/images/52415800-8b69-11e0-9b19-734f5736d2a2"
      }
    ]
  }
}
```

Example 1.11. Image with alternate link: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<image
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  id="52415800-8b69-11e0-9b19-734f5736d2a2"
  name="My Server Backup">
  <atom:link
    rel="self"
    href="http://servers.api.openstack.org/v2/1234/images/52415800-8b69-11e0-9b19-734f5736d2a2"/>
  <atom:link
    rel="bookmark"
    href="http://servers.api.openstack.org/1234/images/52415800-8b69-11e0-9b19-734f5736d2a2"/>
  <atom:link
    rel="alternate" type="application/vnd.openstack.image"
    href="http://glance.api.openstack.org/1234/images/52415800-8b69-11e0-9b19-734f5736d2a2"/>
</image>
```


1.5. Paginated collections

To reduce load on the service, list operations will return a maximum number of items at a time. The maximum number of items returned is determined by the compute provider. To navigate the collection, the parameters *limit* and *marker* can be set in the URI. For example:

```
?limit=100&marker=1234
```

The *marker* parameter is the ID of the last item in the previous list. Items are sorted by create time in descending order. When a create time is not available they are sorted by ID. The *limit* parameter sets the page size. Both parameters are optional. If the client requests a *limit* beyond that which is supported by the deployment an overLimit (413) fault may be thrown. A marker with an invalid ID will return a badRequest (400) fault.

For convenience, collections are required to contain atom "next" links. They may optionally also contain "previous" links. The last page in the list will not contain a "next" link. The following examples illustrate three pages in a collection of images. The first page was retrieved via a **GET** to <http://servers.api.openstack.org/v2/1234/images?limit=1>. In these examples, the *limit* parameter sets the page size to a single item. Subsequent links will honor the initial page size. Thus, a client may follow links to traverse a paginated collection without having to input the *marker* parameter.

Example 1.12. Images collection: XML (first page)

```
<?xml version="1.0" encoding="UTF-8"?>
<images xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <image id="52415800-8b69-11e0-9b19-734f6f006e54" name="CentOS 5.2">
    <atom:link rel="self"
      href="http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"
    />
  </image>
  <atom:link rel="next"
    href="http://servers.api.openstack.org/v2/1234/images?limit=1&
amp;marker=52415800-8b69-11e0-9b19-734f6f006e54"
  />
</images>
```

Example 1.13. Images collection: JSON (first page)

```
{
  "images": [
    {
      "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"
        }
      ]
    }
  ],
  "images_links": [
    {
      "rel": "next",
      "href": "http://servers.api.openstack.org/v2/1234/images?limit=1&
marker=52415800-8b69-11e0-9b19-734f6f006e54"
    }
  ]
}
```

Example 1.14. Images collection: XML (second page)

```
<?xml version="1.0" encoding="UTF-8"?>
<images xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <image id="52415800-8b69-11e0-9b19-734f5736d2a2" name="My Server Backup">
    <atom:link
      rel="self"
      href="http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f5736d2a2"/>
  </image>
  <atom:link rel="next"
    href="http://servers.api.openstack.org/v2/1234/images?limit=1&
amp;marker=52415800-8b69-11e0-9b19-734f5736d2a2"/>
</images>
```

Example 1.15. Images collection: JSON (second page)

```
{
  "images" : [
    {
      "id" : "52415800-8b69-11e0-9b19-734f5736d2a2",
      "name" : "My Server Backup",
      "links": [
        {
          "rel" : "self",
          "href" : "http://servers.api.openstack.org/v2/1234/
images/52415800-8b69-11e0-9b19-734f5736d2a2"
        }
      ]
    }
  ],
  "images_links": [
    {
      "rel" : "next",
      "href" : "http://servers.api.openstack.org/v2/1234/images?limit=1&
marker=52415800-8b69-11e0-9b19-734f5736d2a2"
    }
  ]
}
```

Example 1.16. Images collection: XML (last page)

```
<?xml version="1.0" encoding="UTF-8"?>
<images xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <image id="52415800-8b69-11e0-9b19-734f6ff7c475"
    name="Backup 2">
    <atom:link
      rel="self"
      href="http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6ff7c475"/>
    </image>
  </images>
```

Example 1.17. Images collection: JSON (last page)

```
{
  "images": [
    {
      "id": "52415800-8b69-11e0-9b19-734f6ff7c475",
      "name": "Backup 2",
      "links": [
        {
          "rel": "self",
          "href": "http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6ff7c475"
        }
      ]
    }
  ]
}
```

In JSON, members in a paginated collection are stored in a JSON array named after the collection. A JSON object may also be used to hold members in cases where using an associative array is more practical. Properties about the collection itself, including links, are contained in an array with the name of the entity an underscore (`_`) and `links`. The combination of the objects and arrays that start with the name of the collection and an underscore represent the collection in JSON. The approach allows for extensibility of paginated collections by allowing them to be associated with arbitrary properties. It also allows collections to be embedded in other objects as illustrated below. Here, a subset of metadata items are presented within the image. Clients must follow the "next" link to retrieve the full set of metadata.

Example 1.18. Paginated image metadata: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<image
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  id="52415800-8b69-11e0-9b19-734f6f006e54"
  name="CentOS 5.2">
  <metadata>
    <meta key="ImageVersion">1.5</meta>
    <meta key="ImageType">Gold</meta>
    <atom:link
      rel="next"
      href="http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54/meta?marker=ImageType"/>
    </metadata>
    <atom:link
      rel="self"
      href="http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"/>
  </image>
```

Example 1.19. Paginated image metadata: JSON

```
{
  "image": {
    "id": "52415800-8b69-11e0-9b19-734f6f006e54",
    "name": "CentOS 5.2",
    "metadata": {
      "ImageVersion": "1.5",
      "ImageType": "Gold"
    },
    "metadata_links": [
      {
        "rel": "next",
        "href": "http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54/meta?marker=ImageType"
      }
    ],
    "links": [
      {
        "rel": "self",
        "href": "http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"
      }
    ]
  }
}
```

1.6. Efficient polling with the *Changes-Since* parameter

The ReST API allows you to poll for the status of certain operations by performing a **GET** on various elements. Rather than re-downloading and re-parsing the full status at each polling interval, your ReST client may use the *changes-since* parameter to check for changes since a previous request. The *changes-since* time is specified as an [ISO 8601](#) dateTime (2011-01-24T17:08Z). The form for the timestamp is CCYY-MM-DDThh:mm:ss. An optional time zone may be written in by appending the form ±hh:mm which describes the timezone as an offset from UTC. When the timezone is not specified (2011-01-24T17:08), the UTC timezone will be assumed. If nothing has changed since the *changes-since* time, an empty list will be returned. If data has changed, only the items changed since the specified time will be returned in the response. For example, performing a **GET** against <https://api.servers.openstack.org/v2/224532/servers?changes-since=2011-01-24T17:08Z> would list all servers that have changed since Mon, 24 Jan 2011 17:08:00 UTC.

To allow clients to keep track of changes, the *changes-since* filter displays items that have been *recently* deleted. Both images and servers contain a `DELETED` status that indicates that the resource has been removed. Implementations are not required to keep track of deleted resources indefinitely, so sending a *changes since* time in the distant past may miss deletions.

1.7. Limits

Accounts may be pre-configured with a set of thresholds (or limits) to manage capacity and prevent abuse of the system. The system recognizes two kinds of limits: *rate limits* and *absolute limits*. Rate limits are thresholds that are reset after a certain amount of time passes. Absolute limits are fixed. Limits are configured by operators and may differ from one deployment of the OpenStack Compute service to another. Please contact your provider to determine the limits that apply to your account or see [Section 1.7.3, "Determine limits programmatically" \[20\]](#). Your provider may be able to adjust your account's limits if they are too low.

1.7.1. Rate limits

Rate limits are specified in terms of both a human-readable wild-card URI and a machine-processable regular expression. The human-readable limit is intended for displaying in graphical user interfaces. The machine-processable form is intended to be used directly by client applications.

The regular expression boundary matcher "^" for the rate limit takes effect after the root URI path. For example, the regular expression `^/servers` would match the bolded portion of the following URI: `https://servers.api.openstack.org/v2/3542812/servers`.

Table 1.1. Sample rate limits

Verb	URI	RegEx	Default
POST	*	.*	120/min
POST	<code>*/servers</code>	<code>^/servers</code>	120/min
PUT	*	.*	120/min
GET	<code>*changes-since*</code>	<code>.*changes-since.*</code>	120/min
DELETE	*	.*	120/min
GET	<code>*/os-fping*</code>	<code>^/os-fping</code>	12/min

Rate limits are applied in order relative to the verb, going from least to most specific.

In the event a request exceeds the thresholds established for your account, a 413 HTTP response will be returned with a `Retry-After` header to notify the client when they can attempt to try again.

1.7.2. Absolute limits

Absolute limits are specified as name/value pairs. The name of the absolute limit uniquely identifies the limit within a deployment. Please consult your provider for an exhaustive list of absolute value names. An absolute limit value is always specified as an integer. The name of the absolute limit determines the unit type of the integer value. For example, the name `maxServerMeta` implies that the value is in terms of server metadata items.

Table 1.2. Sample Absolute Limits

Name	Value	Description
<code>maxTotalRAMSize</code>	51200	Maximum total amount of RAM (MB)
<code>maxServerMeta</code>	5	Maximum number of metadata items associated with a server
<code>maxImageMeta</code>	5	Maximum number of metadata items associated with an Image
<code>maxPersonality</code>	5	The maximum number of file path/content pairs that can be supplied on server build
<code>maxPersonalitySize</code>	10240	The maximum size, in bytes, for each personality file

1.7.3. Determine limits programmatically

Applications can programmatically determine current account limits using the `/limits` URI as follows:

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

1.7.3.1. List limits

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

Normal response codes: 200, 203

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

1.7.3.1.1. Request

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

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

This operation does not require a request body.

1.7.3.1.2. Response

Example 1.20. List limits: JSON response

```
{
  "limits": {
    "absolute": {
      "maxImageMeta": 128,
      "maxPersonality": 5,
      "maxPersonalitySize": 10240,
      "maxSecurityGroupRules": 20,
      "maxSecurityGroups": 10,
      "maxServerMeta": 128,
      "maxTotalCores": 20,
      "maxTotalFloatingIps": 10,
      "maxTotalInstances": 10,
      "maxTotalKeypairs": 100,
      "maxTotalRAMSize": 51200
    },
    "rate": [
      {
        "limit": [
          {
            "next-available": "2012-11-27T17:22:18Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "POST"
          },
          {
            "next-available": "2012-11-27T17:22:18Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "PUT"
          }
        ]
      }
    ]
  }
}
```

```

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

```

Example 1.21. List limits: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<limits xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/common/api/v1.0">
  <rates>
    <rate regex=".*" uri="*">

```

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

1.8. Versions

The OpenStack Compute API uses both a URI and a MIME type versioning scheme. In the URI scheme, the first element of the path contains the target version identifier (e.g. `https://servers.api.openstack.org/v1.0/...`). The MIME type versioning scheme uses HTTP content negotiation where the `Accept` or `Content-Type` headers contains a MIME type that identifies the version (`application/vnd.openstack.compute.v2+xml`). A version MIME type is always linked to a base MIME type (`application/xml` or `application/json`). If conflicting versions are specified using both an HTTP header and a URI, the URI takes precedence.

Example 1.22. Request with MIME type versioning

```
GET /214412/images HTTP/1.1
Host: servers.api.openstack.org
Accept: application/vnd.openstack.compute.v2+xml
X-Auth-Token: eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Example 1.23. Request with URI versioning

```
GET /v2/214412/images HTTP/1.1
Host: servers.api.openstack.org
Accept: application/xml
X-Auth-Token: eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```



Note

The MIME type versioning approach allows for the creating of permanent links, because the version scheme is not specified in the URI path: <https://api.servers.openstack.org/224532/servers/123>.

If a request is made without a version specified in the URI or via HTTP headers, then a multiple-choices response (300) will follow providing links and MIME types to available versions.

Example 1.24. Multiple choices: XML response

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

    <atom:link rel="self"
      href="http://servers.api.openstack.org/v1.0/1234/servers/
52415800-8b69-11e0-9b19-734f6af67565"
      />
  </version>

  <version id="v2" status="CURRENT">
    <media-types>
      <media-type base="application/xml"
        type="application/vnd.openstack.compute.v2+xml"/>
      <media-type base="application/json"
        type="application/vnd.openstack.compute.v2+xml"/>
    </media-types>

    <atom:link rel="self"
      href="http://servers.api.openstack.org/v2/1234/servers/
52415800-8b69-11e0-9b19-734f6af67565"
      />
  </version>
</choices>
```

Example 1.25. Multiple choices: JSON response

```
{
  "choices": [
    {
      "id": "v1.0",
      "status": "DEPRECATED",
      "links": [
        {
          "rel": "self",
          "href": "http://servers.api.openstack.org/v1.0/1234/servers/52415800-8b69-11e0-9b19-734f6af67565"
        }
      ],
      "media-types": [
        {
          "base": "application/xml",
          "type": "application/vnd.openstack.compute.v1.0+xml"
        },
        {
          "base": "application/json",
          "type": "application/vnd.openstack.compute.v1.0+json"
        }
      ]
    },
    {
      "id": "v2",
      "status": "CURRENT",
      "links": [
        {
          "rel": "self",
          "href": "http://servers.api.openstack.org/v2/1234/servers/52415800-8b69-11e0-9b19-734f6af67565"
        }
      ],
      "media-types": [
        {
          "base": "application/xml",
          "type": "application/vnd.openstack.compute.v2+xml"
        },
        {
          "base": "application/json",
          "type": "application/vnd.openstack.compute.v2+json"
        }
      ]
    }
  ]
}
```

New features and functionality that do not break API-compatibility will be introduced in the current version of the API as extensions (see below) and the URI and MIME types will remain unchanged. Features or functionality changes that would necessitate a break in API-compatibility will require a new version, which will result in URI and MIME type version being updated accordingly. When new API versions are released, older versions will be marked as `DEPRECATED`. Providers should work with developers and partners to ensure there is adequate time to migrate to the new version before deprecated versions are discontinued.

Your application can programmatically determine available API versions by performing a `GET` on the root URL (i.e. with the version and everything to the right of it truncated) returned from the authentication system. Note that an Atom representation of the versions resources is supported when issuing a request with the `Accept` header containing `application/atom+xml` or by adding a `.atom` to the request URI. This allows standard Atom clients to track version changes.

Example 1.26. List versions: request

```
GET HTTP/1.1
Host: servers.api.openstack.org
```

Normal Response Code(s): 200, 203

Error Response Code(s): 400, 413, 500, 503

This operation does not require a request body.

Example 1.27. List versions: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<versions xmlns="http://docs.openstack.org/common/api/v1.0"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <version id="v1.0" status="DEPRECATED"
    updated="2009-10-09T11:30:00Z">
    <atom:link rel="self"
      href="http://servers.api.openstack.org/v1.0/" />
  </version>
  <version id="v2" status="CURRENT"
    updated="2010-12-12T18:30:02.25Z">
    <atom:link rel="self"
      href="http://servers.api.openstack.org/v2/" />
  </version>
</versions>
```


Example 1.28. List versions: Atom response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title type="text">Available API Versions</title>
  <updated>2010-12-12T18:30:02.25Z</updated>
  <id>http://servers.api.openstack.org/</id>
  <author><name>Rackspace</name><uri>http://www.rackspace.com/</uri></
author>
  <link rel="self" href="http://servers.api.openstack.org/" />
  <entry>
    <id>http://servers.api.openstack.org/v2/</id>
    <title type="text">Version v2</title>
    <updated>2010-12-12T18:30:02.25Z</updated>
    <link rel="self" href="http://servers.api.openstack.org/v2/" />
    <content type="text">Version v2 CURRENT (2010-12-12T18:30:02.25Z)</
content>
  </entry>
  <entry>
    <id>http://servers.api.openstack.org/v1.0/</id>
    <title type="text">Version v1.0</title>
    <updated>2009-10-09T11:30:00Z</updated>
    <link rel="self" href="http://servers.api.openstack.org/v1.0/" />
    <content type="text">Version v1.0 DEPRECATED (2009-10-09T11:30:00Z)</
content>
  </entry>
</feed>
```

Example 1.29. List versions: JSON response

```
{
  "versions": [
    {
      "id": "v1.0",
      "status": "DEPRECATED",
      "updated": "2009-10-09T11:30:00Z",
      "links": [
        {
          "rel": "self",
          "href": "http://servers.api.openstack.org/v1.0/"
        }
      ]
    },
    {
      "id": "v2",
      "status": "CURRENT",
      "updated": "2010-12-12T18:30:02.25Z",
      "links": [
        {
          "rel": "self",
          "href": "http://servers.api.openstack.org/v2/"
        }
      ]
    }
  ]
}
```

You can also obtain additional information about a specific version by performing a **GET** on the base version URL (such as, <https://servers.api.openstack.org/v2/>). Version request URLs should always end with a trailing slash (/). If the slash is omitted, the server may respond with a 302 redirection request. Format extensions may be placed after the slash (e.g. <https://servers.api.openstack.org/v2/.xml>). Note that this is a special case that does not hold true for other API requests. In general, requests such as `/servers.xml` and `/servers/.xml` are handled equivalently.

Example 1.30. Get version details: request

```
GET HTTP/1.1
Host: servers.api.openstack.org/v2/
```

Normal Response Code(s): 200, 203

Error Response Code(s): computeFault (400, 500, ...), serviceUnavailable (503), unauthorized (401), forbidden (403), badRequest (400), badMethod (405), overLimit (413)

This operation does not require a request body

Example 1.31. Get version details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<version xmlns="http://docs.openstack.org/common/api/v1.0"
  xmlns:atom="http://www.w3.org/2005/Atom" id="v2" status="CURRENT"
  updated="2011-01-21T11:33:21-06:00">
  <media-types>
    <media-type base="application/xml"
      type="application/vnd.openstack.compute.v2+xml"/>
    <media-type base="application/json"
      type="application/vnd.openstack.compute.v2+json"/>
  </media-types>
  <atom:link rel="self" href="http://servers.api.openstack.org/v2/" />
  <atom:link rel="describedby" type="application/pdf"
    href="http://docs.rackspacecloud.com/servers/api/v2/cs-
devguide-20110125.pdf" />
  <atom:link rel="describedby" type="application/vnd.sun.wadl+xml"
    href="http://docs.rackspacecloud.com/servers/api/v2/application.wadl"
  />
</version>
```

Example 1.32. Get version details: Atom response

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title type="text">About This Version</title>
  <updated>2011-01-21T11:33:21-06:00</updated>
  <id>http://servers.api.openstack.org/v1.0/</id>
  <author><name>Rackspace</name><uri>http://www.rackspace.com/</uri></author>
  <link rel="self" href="http://servers.api.openstack.org/v1.0/">
  <entry>
    <id>http://servers.api.openstack.org/v1.0/</id>
    <title type="text">Version v2</title>
    <updated>2011-01-21T11:33:21-06:00</updated>
    <link rel="self" href="http://servers.api.openstack.org/v1.0/">
    <link rel="describedby" type="application/pdf"
      href="http://docs.rackspacecloud.com/servers/api/v2/cs-
devguide-20110125.pdf"/>
    <link rel="describedby" type="application/vnd.sun.wadl+xml"
      href="http://docs.rackspacecloud.com/servers/api/v2/application.
wadl"/>
    <content type="text">Version v2 CURRENT (2011-01-21T11:33:21-06:00)</
content>
  </entry>
</feed>
```

Example 1.33. Get version details: JSON response

```
{
  "version" : {
    "id" : "v2",
    "status" : "CURRENT",
    "updated" : "2011-01-21T11:33:21-06:00",
    "links": [
      {
        "rel" : "self",
        "href" : "http://servers.api.openstack.org/v2/"
      },
      {
        "rel" : "describedby",
        "type" : "application/pdf",
        "href" : "http://docs.rackspacecloud.com/servers/api/v2/cs-devguide-20110125.pdf"
      },
      {
        "rel" : "describedby",
        "type" : "application/vnd.sun.wadl+xml",
        "href" : "http://docs.rackspacecloud.com/servers/api/v2/application.wadl"
      }
    ],
    "media-types": [
      {
        "base" : "application/xml",
        "type" : "application/vnd.openstack.compute.v2+xml"
      },
      {
        "base" : "application/json",
        "type" : "application/vnd.openstack.compute.v2+json"
      }
    ]
  }
}
```

The detailed version response contains pointers to both a human-readable and a machine-processable description of the API service. The machine-processable description is written in the Web Application Description Language (WADL).



Note

If a discrepancy exists between the two References, the WADL is authoritative as it contains the most accurate and up-to-date description of the API service.

1.9. Extensions

The OpenStack Compute API is extensible. Extensions serve two purposes: They allow the introduction of new features in the API without requiring a version change and they allow the introduction of vendor specific niche functionality. Applications can programmatically list available extensions by performing a **GET** on the `/extensions` URI. Note that this is a versioned request — that is, an extension available in one API version might not be available in another.

Extensions may also be queried individually by their unique alias. This provides the simplest method of checking if an extension is available as an unavailable extension will issue an `itemNotFound` (404) response.

Extensions may define new data types, parameters, actions, headers, states, and resources. In XML, additional elements and attributes may be defined. These elements must be defined in the extension's namespace. In JSON, the alias must be used. The `volumes` element in the [Examples 1.34 \[32\]](#) and [1.35 \[33\]](#) is defined in the `RS-CBS` namespace. Actions work in exactly the same manner as illustrated in [Examples 1.36 \[35\]](#) and [1.37 \[35\]](#). Extended headers are always prefixed with `X-` followed by the alias and a dash: (`X-RS-CBS-HEADER1`). States and parameters must be prefixed with the extension alias followed by a colon. For example, an image may be in the `RS-PIE:PrepareShare` state.



Important

Applications should be prepared to ignore response data that contains extension elements. An extended state should always be treated as an `UNKNOWN` state if the application does not support the extension. Applications should also verify that an extension is available before submitting an extended request.

Example 1.34. Extended server: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<servers xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <server id="52415800-8b69-11e0-9b19-734f6af67565" tenant_id="1234"
    user_id="5678" name="sample-server" status="BUILD"
    progress="60" hostId="e4d909c290d0fblca068ffaddf22cbd0"
    updated="2010-10-10T12:00:00Z" created="2010-08-10T12:00:00Z"
    accessIPv4="67.23.10.132" accessIPv6="::babe:67.23.10.132">
    <image id="52415800-8b69-11e0-9b19-734f6f006e54">
      <atom:link rel="self"
        href="http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"/>
      <atom:link rel="bookmark"
        href="http://servers.api.openstack.org/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"
      />
    </image>
    <flavor id="52415800-8b69-11e0-9b19-734f216543fd">
      <atom:link rel="self"
        href="http://servers.api.openstack.org/v2/1234/flavors/
52415800-8b69-11e0-9b19-734f216543fd"/>
```

```

        <atom:link rel="bookmark"
            href="http://servers.api.openstack.org/1234/flavors/
52415800-8b69-11e0-9b19-734f216543fd"
        />
    </flavor>
</metadata>
    <meta key="Server Label">Web Head 1</meta>
    <meta key="Image Version">2.1</meta>
</metadata>
<addresses>
    <network id="public">
        <ip version="4" addr="67.23.10.132"/>
        <ip version="6" addr="::babe:67.23.10.132"/>
        <ip version="4" addr="67.23.10.131"/>
        <ip version="6" addr="::babe:4317:0A83"/>
    </network>
    <network id="private">
        <ip version="4" addr="10.176.42.16"/>
        <ip version="6" addr="::babe:10.176.42.16"/>
    </network>
</addresses>
    <atom:link rel="self"
        href="http://servers.api.openstack.org/v2/1234/servers/
52415800-8b69-11e0-9b19-734f6af67565"/>
    <atom:link rel="bookmark"
        href="http://servers.api.openstack.org/1234/servers/
52415800-8b69-11e0-9b19-734f6af67565"/>
    <volumes
        xmlns="http://docs.rackspacecloud.com/servers/api/ext/cbs/v1.
0">
        <volume name="OS"
            href="https://cbs.api.rackspacecloud.com/12934/volumes/
19"/>
        <volume name="Work"
            href="https://cbs.api.rackspacecloud.com/12934/volumes/23"
        />
    </volumes>
</server>
</servers>

```

Example 1.35. Extended server: JSON response

```

{
  "servers": [
    {
      "id": "52415800-8b69-11e0-9b19-734f6af67565",
      "tenant_id": "1234",
      "user_id": "5678",
      "name": "sample-server",
      "updated": "2010-10-10T12:00:00Z",
      "created": "2010-08-10T12:00:00Z",
      "hostId": "e4d909c290d0fblca068ffaddf22cbd0",
      "status": "BUILD",
      "progress": 60,
      "accessIPv4": "67.23.10.132",
      "accessIPv6": "::babe:67.23.10.132",
      "image": {

```

```

        "id": "52415800-8b69-11e0-9b19-734f6f006e54",
        "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": "52415800-8b69-11e0-9b19-734f216543fd",
        "links": [
            {
                "rel": "self",
                "href": "http://servers.api.openstack.org/v2/1234/
flavors/52415800-8b69-11e0-9b19-734f216543fd"
            },
            {
                "rel": "bookmark",
                "href": "http://servers.api.openstack.org/1234/
flavors/52415800-8b69-11e0-9b19-734f216543fd"
            }
        ]
    },
    "addresses": {
        "public": [
            {
                "version": 4,
                "addr": "67.23.10.132"
            },
            {
                "version": 6,
                "addr": "::babe:67.23.10.132"
            },
            {
                "version": 4,
                "addr": "67.23.10.131"
            },
            {
                "version": 6,
                "addr": "::babe:4317:0A83"
            }
        ],
        "private": [
            {
                "version": 4,
                "addr": "10.176.42.16"
            },
            {
                "version": 6,
                "addr": "::babe:10.176.42.16"
            }
        ]
    }
}

```

```

    ],
    "metadata": {
      "Server Label": "Web Head 1",
      "Image Version": "2.1"
    },
    "links": [
      {
        "rel": "self",
        "href": "http://servers.api.openstack.org/v2/1234/servers/52415800-8b69-11e0-9b19-734f6af67565"
      },
      {
        "rel": "bookmark",
        "href": "http://servers.api.openstack.org/1234/servers/52415800-8b69-11e0-9b19-734f6af67565"
      }
    ],
    "RS-CBS:volumes": [
      {
        "name": "OS",
        "href": "https://cbs.api.rackspacecloud.com/12934/volumes/19"
      },
      {
        "name": "Work",
        "href": "https://cbs.api.rackspacecloud.com/12934/volumes/23"
      }
    ]
  }
}

```

Example 1.36. Extended action: XML response

```

<?xml version="1.0" encoding="UTF-8"?>
<attach-volume
  xmlns="http://docs.rackspacecloud.com/servers/api/ext/cbs/v1.0"
  href="https://cbs.api.rackspacecloud.com/12934/volumes/19" />

```

Example 1.37. Extended action: JSON response

```

{
  "RS-CBS:attach-volume": {
    "href": "https://cbs.api.rackspacecloud.com/12934/volumes/19"
  }
}

```

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

1.9.1. List extensions

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

Normal response codes: 200, 203

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

1.9.1.1. Request

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

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

This operation does not require a request body.

1.9.1.2. Response

Example 1.38. List extensions: JSON response

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

```

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

```

```

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

```

```

        "namespace": "http://docs.openstack.org/compute/ext/assisted-
volume-snapshots/api/v2",
        "updated": "2013-08-15T00:00:00-00:00"
    },
    {
        "alias": "os-attach-interfaces",
        "description": "Attach interface support.",
        "links": [],
        "name": "AttachInterfaces",
        "namespace": "http://docs.openstack.org/compute/ext/interfaces/
api/v1.1",
        "updated": "2012-07-22T00:00:00+00:00"
    },
    {
        "alias": "os-availability-zone",
        "description": "1. Add availability_zone to the Create Server v1.1
API.\n
        2. Add availability zones describing.\n
        ",
        "links": [],
        "name": "AvailabilityZone",
        "namespace": "http://docs.openstack.org/compute/ext/
availabilityzone/api/v1.1",
        "updated": "2012-12-21T00:00:00+00:00"
    },
    {
        "alias": "os-baremetal-nodes",
        "description": "Admin-only bare-metal node administration.",
        "links": [],
        "name": "BareMetalNodes",
        "namespace": "http://docs.openstack.org/compute/ext/
baremetal_nodes/api/v2",
        "updated": "2013-01-04T00:00:00+00:00"
    },
    {
        "alias": "os-block-device-mapping-v2-boot",
        "description": "Allow boot with the new BDM data format.",
        "links": [],
        "name": "BlockDeviceMappingV2Boot",
        "namespace": "http://docs.openstack.org/compute/ext/
block_device_mapping_v2_boot/api/v2",
        "updated": "2013-07-08T00:00:00+00:00"
    },
    {
        "alias": "os-cell-capacities",
        "description": "Adding functionality to get cell capacities.",
        "links": [],
        "name": "CellCapacities",
        "namespace": "http://docs.openstack.org/compute/ext/
cell_capacities/api/v1.1",
        "updated": "2013-05-27T00:00:00+00:00"
    },
    {
        "alias": "os-cells",
        "description": "Enables cells-related functionality such as adding
neighbor cells,\n
        listing neighbor cells, and getting the capabilities of
the local cell.\n
        ",
        "links": [],
        "name": "Cells",
        "namespace": "http://docs.openstack.org/compute/ext/cells/api/v1.
1",
        "updated": "2013-05-14T00:00:00+00:00"
    }

```

```

    },
    {
      "alias": "os-certificates",
      "description": "Certificates support.",
      "links": [],
      "name": "Certificates",
      "namespace": "http://docs.openstack.org/compute/ext/certificates/
api/v1.1",
      "updated": "2012-01-19T00:00:00+00:00"
    },
    {
      "alias": "os-cloudpipe",
      "description": "Adds actions to create cloudpipe instances.\n\n
      When running with the Vlan network mode, you need a mechanism to route\
      n from the public Internet to your vlans. This mechanism is known as a\
      n cloudpipe.\n\n At the time of creating this class, only OpenVPN is
      supported. Support for\n a SSH Bastion host is forthcoming.\n ",
      "links": [],
      "name": "Cloudpipe",
      "namespace": "http://docs.openstack.org/compute/ext/cloudpipe/api/
v1.1",
      "updated": "2011-12-16T00:00:00+00:00"
    },
    {
      "alias": "os-cloudpipe-update",
      "description": "Adds the ability to set the vpn ip/port for
      cloudpipe instances.",
      "links": [],
      "name": "CloudpipeUpdate",
      "namespace": "http://docs.openstack.org/compute/ext/cloudpipe-
update/api/v2",
      "updated": "2012-11-14T00:00:00+00:00"
    },
    {
      "alias": "os-config-drive",
      "description": "Config Drive Extension.",
      "links": [],
      "name": "ConfigDrive",
      "namespace": "http://docs.openstack.org/compute/ext/config_drive/
api/v1.1",
      "updated": "2012-07-16T00:00:00+00:00"
    },
    {
      "alias": "os-console-output",
      "description": "Console log output support, with tailing ability.
      ",
      "links": [],
      "name": "ConsoleOutput",
      "namespace": "http://docs.openstack.org/compute/ext/os-console-
output/api/v2",
      "updated": "2011-12-08T00:00:00+00:00"
    },
    {
      "alias": "os-consoles",
      "description": "Interactive Console support.",
      "links": [],
      "name": "Consoles",
      "namespace": "http://docs.openstack.org/compute/ext/os-consoles/
api/v2",
      "updated": "2011-12-23T00:00:00+00:00"
    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "http://docs.openstack.org/compute/ext/os-certificates/
api/v1.1"
    },
    {
      "rel": "self",
      "href": "http://docs.openstack.org/compute/ext/cloudpipe/api/
v1.1"
    },
    {
      "rel": "self",
      "href": "http://docs.openstack.org/compute/ext/cloudpipe-
update/api/v2"
    },
    {
      "rel": "self",
      "href": "http://docs.openstack.org/compute/ext/config_drive/
api/v1.1"
    },
    {
      "rel": "self",
      "href": "http://docs.openstack.org/compute/ext/os-console-
output/api/v2"
    },
    {
      "rel": "self",
      "href": "http://docs.openstack.org/compute/ext/os-consoles/
api/v2"
    }
  ],
  "name": "os-extensions",
  "namespace": "http://docs.openstack.org/compute/ext/os-extensions/
api/v2",
  "updated": "2011-12-23T00:00:00+00:00"
}

```

```

    },
    {
      "alias": "os-coverage",
      "description": "Enable Nova Coverage.",
      "links": [],
      "name": "Coverage",
      "namespace": "http://docs.openstack.org/compute/ext/coverage/api/
v2",
      "updated": "2012-10-15T00:00:00+00:00"
    },
    {
      "alias": "os-create-server-ext",
      "description": "Extended support to the Create Server v1.1 API.",
      "links": [],
      "name": "Createserverext",
      "namespace": "http://docs.openstack.org/compute/ext/
createserverext/api/v1.1",
      "updated": "2011-07-19T00:00:00+00:00"
    },
    {
      "alias": "os-deferred-delete",
      "description": "Instance deferred delete.",
      "links": [],
      "name": "DeferredDelete",
      "namespace": "http://docs.openstack.org/compute/ext/deferred-
delete/api/v1.1",
      "updated": "2011-09-01T00:00:00+00:00"
    },
    {
      "alias": "os-evacuate",
      "description": "Enables server evacuation.",
      "links": [],
      "name": "Evacuate",
      "namespace": "http://docs.openstack.org/compute/ext/evacuate/api/
v2",
      "updated": "2013-01-06T00:00:00+00:00"
    },
    {
      "alias": "os-extended-floating-ips",
      "description": "Adds optional fixed_address to the add floating IP
command.",
      "links": [],
      "name": "ExtendedFloatingIps",
      "namespace": "http://docs.openstack.org/compute/ext/
extended_floating_ips/api/v2",
      "updated": "2013-04-19T00:00:00+00:00"
    },
    {
      "alias": "os-extended-quotas",
      "description": "Adds ability for admins to delete quota\n and
optionally force the update Quota command.\n ",
      "links": [],
      "name": "ExtendedQuotas",
      "namespace": "http://docs.openstack.org/compute/ext/
extended_quotas/api/v1.1",
      "updated": "2013-06-09T00:00:00+00:00"
    },
    {
      "alias": "os-extended-services",
      "description": "Extended services support.",

```

```

        "links": [],
        "name": "ExtendedServices",
        "namespace": "http://docs.openstack.org/compute/ext/
extended_services/api/v2",
        "updated": "2013-05-17T00:00:00-00:00"
    },
    {
        "alias": "os-extended-volumes",
        "description": "Extended Volumes support.",
        "links": [],
        "name": "ExtendedVolumes",
        "namespace": "http://docs.openstack.org/compute/ext/
extended_volumes/api/v1.1",
        "updated": "2013-06-07T00:00:00+00:00"
    },
    {
        "alias": "os-fixed-ips",
        "description": "Fixed IPs support.",
        "links": [],
        "name": "FixedIPs",
        "namespace": "http://docs.openstack.org/compute/ext/fixed_ips/api/
v2",
        "updated": "2012-10-18T13:25:27-06:00"
    },
    {
        "alias": "os-flavor-access",
        "description": "Flavor access support.",
        "links": [],
        "name": "FlavorAccess",
        "namespace": "http://docs.openstack.org/compute/ext/flavor_access/
api/v2",
        "updated": "2012-08-01T00:00:00+00:00"
    },
    {
        "alias": "os-flavor-extra-specs",
        "description": "Instance type (flavor) extra specs.",
        "links": [],
        "name": "FlavorExtraSpecs",
        "namespace": "http://docs.openstack.org/compute/ext/
flavor_extra_specs/api/v1.1",
        "updated": "2011-06-23T00:00:00+00:00"
    },
    {
        "alias": "os-flavor-manage",
        "description": "\n    Flavor create/delete API support\n    ",
        "links": [],
        "name": "FlavorManage",
        "namespace": "http://docs.openstack.org/compute/ext/flavor_manage/
api/v1.1",
        "updated": "2012-01-19T00:00:00+00:00"
    },
    {
        "alias": "os-flavor-rxtx",
        "description": "Support to show the rxtx status of a flavor.",
        "links": [],
        "name": "FlavorRxtx",
        "namespace": "http://docs.openstack.org/compute/ext/flavor_rxtx/
api/v1.1",
        "updated": "2012-08-29T00:00:00+00:00"
    },

```

```

        {
            "alias": "os-flavor-swap",
            "description": "Support to show the swap status of a flavor.",
            "links": [],
            "name": "FlavorSwap",
            "namespace": "http://docs.openstack.org/compute/ext/flavor_swap/
api/v1.1",
            "updated": "2012-08-29T00:00:00+00:00"
        },
        {
            "alias": "os-floating-ip-dns",
            "description": "Floating IP DNS support.",
            "links": [],
            "name": "FloatingIpDns",
            "namespace": "http://docs.openstack.org/ext/floating_ip_dns/api/
v1.1",
            "updated": "2011-12-23T00:00:00+00:00"
        },
        {
            "alias": "os-floating-ip-pools",
            "description": "Floating IPs support.",
            "links": [],
            "name": "FloatingIpPools",
            "namespace": "http://docs.openstack.org/compute/ext/
floating_ip_pools/api/v1.1",
            "updated": "2012-01-04T00:00:00+00:00"
        },
        {
            "alias": "os-floating-ips",
            "description": "Floating IPs support.",
            "links": [],
            "name": "FloatingIps",
            "namespace": "http://docs.openstack.org/compute/ext/floating_ips/
api/v1.1",
            "updated": "2011-06-16T00:00:00+00:00"
        },
        {
            "alias": "os-floating-ips-bulk",
            "description": "Bulk handling of Floating IPs.",
            "links": [],
            "name": "FloatingIpsBulk",
            "namespace": "http://docs.openstack.org/compute/ext/
floating_ips_bulk/api/v2",
            "updated": "2012-10-29T13:25:27-06:00"
        },
        {
            "alias": "os-fping",
            "description": "Fping Management Extension.",
            "links": [],
            "name": "Fping",
            "namespace": "http://docs.openstack.org/compute/ext/fping/api/v1.
1",
            "updated": "2012-07-06T00:00:00+00:00"
        },
        {
            "alias": "os-hide-server-addresses",
            "description": "Support hiding server addresses in certain states.
",
            "links": [],
            "name": "HideServerAddresses",

```



```

        "namespace": "http://docs.openstack.org/compute/ext/
hide_server_addresses/api/v1.1",
        "updated": "2012-12-11T00:00:00+00:00"
    },
    {
        "alias": "os-hosts",
        "description": "Admin-only host administration.",
        "links": [],
        "name": "Hosts",
        "namespace": "http://docs.openstack.org/compute/ext/hosts/api/v1.
1",
        "updated": "2011-06-29T00:00:00+00:00"
    },
    {
        "alias": "os-hypervisors",
        "description": "Admin-only hypervisor administration.",
        "links": [],
        "name": "Hypervisors",
        "namespace": "http://docs.openstack.org/compute/ext/hypervisors/
api/v1.1",
        "updated": "2012-06-21T00:00:00+00:00"
    },
    {
        "alias": "os-instance-actions",
        "description": "View a log of actions and events taken on an
instance.",
        "links": [],
        "name": "InstanceActions",
        "namespace": "http://docs.openstack.org/compute/ext/instance-
actions/api/v1.1",
        "updated": "2013-02-08T00:00:00+00:00"
    },
    {
        "alias": "os-instance_usage_audit_log",
        "description": "Admin-only Task Log Monitoring.",
        "links": [],
        "name": "OSInstanceUsageAuditLog",
        "namespace": "http://docs.openstack.org/ext/services/api/v1.1",
        "updated": "2012-07-06T01:00:00+00:00"
    },
    {
        "alias": "os-keypairs",
        "description": "Keypair Support.",
        "links": [],
        "name": "Keypairs",
        "namespace": "http://docs.openstack.org/compute/ext/keypairs/api/
v1.1",
        "updated": "2011-08-08T00:00:00+00:00"
    },
    {
        "alias": "os-migrations",
        "description": "Provide data on migrations.",
        "links": [],
        "name": "Migrations",
        "namespace": "http://docs.openstack.org/compute/ext/migrations/
api/v2.0",
        "updated": "2013-05-30T00:00:00+00:00"
    },
    {
        "alias": "os-multiple-create",

```

```

        "description": "Allow multiple create in the Create Server v1.1
API.",
        "links": [],
        "name": "MultipleCreate",
        "namespace": "http://docs.openstack.org/compute/ext/
multiplecreate/api/v1.1",
        "updated": "2012-08-07T00:00:00+00:00"
    },
    {
        "alias": "os-networks",
        "description": "Admin-only Network Management Extension.",
        "links": [],
        "name": "Networks",
        "namespace": "http://docs.openstack.org/compute/ext/os-networks/
api/v1.1",
        "updated": "2011-12-23T00:00:00+00:00"
    },
    {
        "alias": "os-networks-associate",
        "description": "Network association support.",
        "links": [],
        "name": "NetworkAssociationSupport",
        "namespace": "http://docs.openstack.org/compute/ext/
networks_associate/api/v2",
        "updated": "2012-11-19T00:00:00+00:00"
    },
    {
        "alias": "os-quota-class-sets",
        "description": "Quota classes management support.",
        "links": [],
        "name": "QuotaClasses",
        "namespace": "http://docs.openstack.org/compute/ext/quota-classes-
sets/api/v1.1",
        "updated": "2012-03-12T00:00:00+00:00"
    },
    {
        "alias": "os-quota-sets",
        "description": "Quotas management support.",
        "links": [],
        "name": "Quotas",
        "namespace": "http://docs.openstack.org/compute/ext/quotas-sets/
api/v1.1",
        "updated": "2011-08-08T00:00:00+00:00"
    },
    {
        "alias": "os-rescue",
        "description": "Instance rescue mode.",
        "links": [],
        "name": "Rescue",
        "namespace": "http://docs.openstack.org/compute/ext/rescue/api/v1.
1",
        "updated": "2011-08-18T00:00:00+00:00"
    },
    {
        "alias": "os-security-group-default-rules",
        "description": "Default rules for security group support.",
        "links": [],
        "name": "SecurityGroupDefaultRules",
        "namespace": "http://docs.openstack.org/compute/ext/
securitygroupdefaultrules/api/v1.1",

```

```

        "updated": "2013-02-05T00:00:00+00:00"
    },
    {
        "alias": "os-security-groups",
        "description": "Security group support.",
        "links": [],
        "name": "SecurityGroups",
        "namespace": "http://docs.openstack.org/compute/ext/
securitygroups/api/v1.1",
        "updated": "2013-05-28T00:00:00+00:00"
    },
    {
        "alias": "os-server-diagnostics",
        "description": "Allow Admins to view server diagnostics through
server action.",
        "links": [],
        "name": "ServerDiagnostics",
        "namespace": "http://docs.openstack.org/compute/ext/server-
diagnostics/api/v1.1",
        "updated": "2011-12-21T00:00:00+00:00"
    },
    {
        "alias": "os-server-password",
        "description": "Server password support.",
        "links": [],
        "name": "ServerPassword",
        "namespace": "http://docs.openstack.org/compute/ext/server-
password/api/v2",
        "updated": "2012-11-29T00:00:00+00:00"
    },
    {
        "alias": "os-server-start-stop",
        "description": "Start/Stop instance compute API support.",
        "links": [],
        "name": "ServerStartStop",
        "namespace": "http://docs.openstack.org/compute/ext/servers/api/
v1.1",
        "updated": "2012-01-23T00:00:00+00:00"
    },
    {
        "alias": "os-services",
        "description": "Services support.",
        "links": [],
        "name": "Services",
        "namespace": "http://docs.openstack.org/compute/ext/services/api/
v2",
        "updated": "2012-10-28T00:00:00-00:00"
    },
    {
        "alias": "os-shelve",
        "description": "Instance shelve mode.",
        "links": [],
        "name": "Shelve",
        "namespace": "http://docs.openstack.org/compute/ext/shelve/api/v1.
1",
        "updated": "2013-04-06T00:00:00+00:00"
    },
    {
        "alias": "os-simple-tenant-usage",
        "description": "Simple tenant usage extension.",

```

```

        "links": [],
        "name": "SimpleTenantUsage",
        "namespace": "http://docs.openstack.org/compute/ext/os-simple-tenant-usage/api/v1.1",
        "updated": "2011-08-19T00:00:00+00:00"
    },
    {
        "alias": "os-tenant-networks",
        "description": "Tenant-based Network Management Extension.",
        "links": [],
        "name": "OSTenantNetworks",
        "namespace": "http://docs.openstack.org/compute/ext/os-tenant-networks/api/v2",
        "updated": "2012-03-07T09:46:43-05:00"
    },
    {
        "alias": "os-used-limits",
        "description": "Provide data on limited resources that are being used.",
        "links": [],
        "name": "UsedLimits",
        "namespace": "http://docs.openstack.org/compute/ext/used_limits/api/v1.1",
        "updated": "2012-07-13T00:00:00+00:00"
    },
    {
        "alias": "os-used-limits-for-admin",
        "description": "Provide data to admin on limited resources used by other tenants.",
        "links": [],
        "name": "UsedLimitsForAdmin",
        "namespace": "http://docs.openstack.org/compute/ext/used_limits_for_admin/api/v1.1",
        "updated": "2013-05-02T00:00:00+00:00"
    },
    {
        "alias": "os-user-data",
        "description": "Add user_data to the Create Server v1.1 API.",
        "links": [],
        "name": "UserData",
        "namespace": "http://docs.openstack.org/compute/ext/userdata/api/v1.1",
        "updated": "2012-08-07T00:00:00+00:00"
    },
    {
        "alias": "os-user-quotas",
        "description": "Project user quota support.",
        "links": [],
        "name": "UserQuotas",
        "namespace": "http://docs.openstack.org/compute/ext/user_quotas/api/v1.1",
        "updated": "2013-07-18T00:00:00+00:00"
    },
    {
        "alias": "os-virtual-interfaces",
        "description": "Virtual interface support.",
        "links": [],
        "name": "VirtualInterfaces",
        "namespace": "http://docs.openstack.org/compute/ext/virtual_interfaces/api/v1.1",

```

```

        "updated": "2011-08-17T00:00:00+00:00"
      },
      {
        "alias": "os-volume-attachment-update",
        "description": "Support for updating a volume attachment.",
        "links": [],
        "name": "VolumeAttachmentUpdate",
        "namespace": "http://docs.openstack.org/compute/ext/os-volume-attachment-update/api/v2",
        "updated": "2013-06-20T00:00:00-00:00"
      },
      {
        "alias": "os-volumes",
        "description": "Volumes support.",
        "links": [],
        "name": "Volumes",
        "namespace": "http://docs.openstack.org/compute/ext/volumes/api/v1.1",
        "updated": "2011-03-25T00:00:00+00:00"
      }
    ]
  }
}

```

Example 1.39. List extensions: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<extensions xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/common/api/v1.0">
  <extension alias="NMN" updated="2011-06-09T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/multinic/api/v1.1" name="Multinic">
    <description>Multiple network support.</description>
  </extension>
  <extension alias="OS-DCF" updated="2011-09-27T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/disk_config/api/v1.1" name="DiskConfig">
    <description>Disk Management Extension.</description>
  </extension>
  <extension alias="OS-EXT-AZ" updated="2013-01-30T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/extended_availability_zone/api/v2" name="ExtendedAvailabilityZone">
    <description>Extended Server Attributes support.</description>
  </extension>
  <extension alias="OS-EXT-IMG-SIZE" updated="2013-02-19T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/image_size/api/v1.1" name="ImageSize">
    <description>Adds image size to image listings.</description>
  </extension>
  <extension alias="OS-EXT-IPS" updated="2013-01-06T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/extended_ips/api/v1.1" name="ExtendedIps">
    <description>Adds type parameter to the ip list.</description>
  </extension>
  <extension alias="OS-EXT-IPS-MAC" updated="2013-03-07T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/extended_ips_mac/api/v1.1" name="ExtendedIpsMac">
    <description>Adds mac address parameter to the ip list.</description>
  </extension>
  <extension alias="OS-EXT-SRV-ATTR" updated="2011-11-03T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/extended_status/api/v1.1" name="ExtendedServerAttributes">

```

```

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

    Actions include: pause, unpause, suspend, resume, migrate,
    resetNetwork, injectNetworkInfo, lock, unlock, createBackup
  </description>
  </extension>
  <extension alias="os-agents" updated="2012-10-28T00:00:00-00:00" namespace=
"http://docs.openstack.org/compute/ext/agents/api/v2" name="Agents">
    <description>Agents support.</description>
  </extension>
  <extension alias="os-aggregates" updated="2012-01-12T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/aggregates/api/v1.1" name=
"Aggregates">
    <description>Admin-only aggregate administration.</description>
  </extension>
  <extension alias="os-assisted-volume-snapshots" updated=
"2013-08-15T00:00:00-00:00" namespace="http://docs.openstack.org/compute/ext/
assisted-volume-snapshots/api/v2" name="AssistedVolumeSnapshots">
    <description>Assisted volume snapshots.</description>
  </extension>

```

```

<extension alias="os-attach-interfaces" updated="2012-07-22T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/interfaces/api/v1.1" name=
"AttachInterfaces">
  <description>Attach interface support.</description>
</extension>
<extension alias="os-availability-zone" updated="2012-12-21T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/availabilityzone/api/v1.1"
name="AvailabilityZone">
  <description>1. Add availability_zone to the Create Server v1.1 API.
  2. Add availability zones describing.
</description>
</extension>
<extension alias="os-baremetal-nodes" updated="2013-01-04T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/baremetal_nodes/api/v2"
name="BareMetalNodes">
  <description>Admin-only bare-metal node administration.</description>
</extension>
<extension alias="os-block-device-mapping-v2-boot" updated=
"2013-07-08T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/
block_device_mapping_v2_boot/api/v2" name="BlockDeviceMappingV2Boot">
  <description>Allow boot with the new BDM data format.</description>
</extension>
<extension alias="os-cell-capacities" updated="2013-05-27T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/cell_capacities/api/v1.1"
name="CellCapacities">
  <description>Adding functionality to get cell capacities.</description>
</extension>
<extension alias="os-cells" updated="2013-05-14T00:00:00+00:00" namespace=
"http://docs.openstack.org/compute/ext/cells/api/v1.1" name="Cells">
  <description>Enables cells-related functionality such as adding neighbor
cells,
  listing neighbor cells, and getting the capabilities of the local cell.
</description>
</extension>
<extension alias="os-certificates" updated="2012-01-19T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/certificates/api/v1.1" name=
"Certificates">
  <description>Certificates support.</description>
</extension>
<extension alias="os-cloudpipe" updated="2011-12-16T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/cloudpipe/api/v1.1" name=
"Cloudpipe">
  <description>Adds actions to create cloudpipe instances.

  When running with the Vlan network mode, you need a mechanism to route
  from the public Internet to your vlans. This mechanism is known as a
  cloudpipe.

  At the time of creating this class, only OpenVPN is supported. Support
  for
  a SSH Bastion host is forthcoming.
</description>
</extension>
<extension alias="os-cloudpipe-update" updated="2012-11-14T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/cloudpipe-update/api/v2"
name="CloudpipeUpdate">
  <description>Adds the ability to set the vpn ip/port for cloudpipe
instances.</description>
</extension>

```

```

    <extension alias="os-config-drive" updated="2012-07-16T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/config_drive/api/v1.1" name=
    "ConfigDrive">
      <description>Config Drive Extension.</description>
    </extension>
    <extension alias="os-console-output" updated="2011-12-08T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/os-console-output/api/v2"
    name="ConsoleOutput">
      <description>Console log output support, with tailing ability.</
    description>
    </extension>
    <extension alias="os-consoles" updated="2011-12-23T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/os-consoles/api/v2" name=
    "Consoles">
      <description>Interactive Console support.</description>
    </extension>
    <extension alias="os-coverage" updated="2012-10-15T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/coverage/api/v2" name=
    "Coverage">
      <description>Enable Nova Coverage.</description>
    </extension>
    <extension alias="os-create-server-ext" updated="2011-07-19T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/createserverext/api/v1.1"
    name="Createserverext">
      <description>Extended support to the Create Server v1.1 API.</description>
    </extension>
    <extension alias="os-deferred-delete" updated="2011-09-01T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/deferred-delete/api/v1.1"
    name="DeferredDelete">
      <description>Instance deferred delete.</description>
    </extension>
    <extension alias="os-evacuate" updated="2013-01-06T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/evacuate/api/v2" name=
    "Evacuate">
      <description>Enables server evacuation.</description>
    </extension>
    <extension alias="os-extended-floating-ips" updated=
    "2013-04-19T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/
    extended_floating_ips/api/v2" name="ExtendedFloatingIps">
      <description>Adds optional fixed_address to the add floating IP command.</
    description>
    </extension>
    <extension alias="os-extended-quotas" updated="2013-06-09T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_quotas/api/v1.1"
    name="ExtendedQuotas">
      <description>Adds ability for admins to delete quota
      and optionally force the update Quota command.
    </description>
    </extension>
    <extension alias="os-extended-services" updated="2013-05-17T00:00:00-00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_services/api/v2"
    name="ExtendedServices">
      <description>Extended services support.</description>
    </extension>
    <extension alias="os-extended-volumes" updated="2013-06-07T00:00:00+00:00"
    namespace="http://docs.openstack.org/compute/ext/extended_volumes/api/v1.1"
    name="ExtendedVolumes">
      <description>Extended Volumes support.</description>
    </extension>

```



```

<extension alias="os-fixed-ips" updated="2012-10-18T13:25:27-06:00"
namespace="http://docs.openstack.org/compute/ext/fixed_ips/api/v2" name=
"FixedIPs">
  <description>Fixed IPs support.</description>
</extension>
<extension alias="os-flavor-access" updated="2012-08-01T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/flavor_access/api/v2" name=
"FlavorAccess">
  <description>Flavor access support.</description>
</extension>
<extension alias="os-flavor-extra-specs" updated="2011-06-23T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/flavor_extra_specs/api/v1.1"
name="FlavorExtraSpecs">
  <description>Instance type (flavor) extra specs.</description>
</extension>
<extension alias="os-flavor-manage" updated="2012-01-19T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/flavor_manage/api/v1.1"
name="FlavorManage">
  <description>
    Flavor create/delete API support
  </description>
</extension>
<extension alias="os-flavor-rxtx" updated="2012-08-29T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/flavor_rxtx/api/v1.1" name=
"FlavorRxtx">
  <description>Support to show the rxtx status of a flavor.</description>
</extension>
<extension alias="os-flavor-swap" updated="2012-08-29T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/flavor_swap/api/v1.1" name=
"FlavorSwap">
  <description>Support to show the swap status of a flavor.</description>
</extension>
<extension alias="os-floating-ip-dns" updated="2011-12-23T00:00:00+00:00"
namespace="http://docs.openstack.org/ext/floating_ip_dns/api/v1.1" name=
"FloatingIpDns">
  <description>Floating IP DNS support.</description>
</extension>
<extension alias="os-floating-ip-pools" updated="2012-01-04T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/floating_ip_pools/api/v1.1"
name="FloatingIpPools">
  <description>Floating IPs support.</description>
</extension>
<extension alias="os-floating-ips" updated="2011-06-16T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/floating_ips/api/v1.1" name=
"FloatingIps">
  <description>Floating IPs support.</description>
</extension>
<extension alias="os-floating-ips-bulk" updated="2012-10-29T13:25:27-06:00"
namespace="http://docs.openstack.org/compute/ext/floating_ips_bulk/api/v2"
name="FloatingIpsBulk">
  <description>Bulk handling of Floating IPs.</description>
</extension>
<extension alias="os-fping" updated="2012-07-06T00:00:00+00:00" namespace=
"http://docs.openstack.org/compute/ext/fping/api/v1.1" name="Fping">
  <description>Fping Management Extension.</description>
</extension>
<extension alias="os-hide-server-addresses" updated=
"2012-12-11T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/
hide_server_addresses/api/v1.1" name="HideServerAddresses">

```

```

    <description>Support hiding server addresses in certain states.</
description>
</extension>
<extension alias="os-hosts" updated="2011-06-29T00:00:00+00:00" namespace=
"http://docs.openstack.org/compute/ext/hosts/api/v1.1" name="Hosts">
    <description>Admin-only host administration.</description>
</extension>
<extension alias="os-hypervisors" updated="2012-06-21T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/hypervisors/api/v1.1" name=
"Hypervisors">
    <description>Admin-only hypervisor administration.</description>
</extension>
<extension alias="os-instance-actions" updated="2013-02-08T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/instance-actions/api/v1.1"
name="InstanceActions">
    <description>View a log of actions and events taken on an instance.</
description>
</extension>
<extension alias="os-instance_usage_audit_log" updated=
"2012-07-06T01:00:00+00:00" namespace="http://docs.openstack.org/ext/services/
api/v1.1" name="OSInstanceUsageAuditLog">
    <description>Admin-only Task Log Monitoring.</description>
</extension>
<extension alias="os-keypairs" updated="2011-08-08T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/keypairs/api/v1.1" name=
"Keypairs">
    <description>Keypair Support.</description>
</extension>
<extension alias="os-migrations" updated="2013-05-30T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/migrations/api/v2.0" name=
"Migrations">
    <description>Provide data on migrations.</description>
</extension>
<extension alias="os-multiple-create" updated="2012-08-07T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/multiplecreate/api/v1.1"
name="MultipleCreate">
    <description>Allow multiple create in the Create Server v1.1 API.</
description>
</extension>
<extension alias="os-networks" updated="2011-12-23T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/os-networks/api/v1.1" name=
"Networks">
    <description>Admin-only Network Management Extension.</description>
</extension>
<extension alias="os-networks-associate" updated="2012-11-19T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/networks_associate/api/v2"
name="NetworkAssociationSupport">
    <description>Network association support.</description>
</extension>
<extension alias="os-quota-class-sets" updated="2012-03-12T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/quota-classes-sets/api/v1.1"
name="QuotaClasses">
    <description>Quota classes management support.</description>
</extension>
<extension alias="os-quota-sets" updated="2011-08-08T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/quotas-sets/api/v1.1" name=
"Quotas">
    <description>Quotas management support.</description>
</extension>

```

```

<extension alias="os-rescue" updated="2011-08-18T00:00:00+00:00" namespace=
"http://docs.openstack.org/compute/ext/rescue/api/v1.1" name="Rescue">
  <description>Instance rescue mode.</description>
</extension>
<extension alias="os-security-group-default-rules" updated=
"2013-02-05T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/
securitygroupdefaultrules/api/v1.1" name="SecurityGroupDefaultRules">
  <description>Default rules for security group support.</description>
</extension>
<extension alias="os-security-groups" updated="2013-05-28T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/securitygroups/api/v1.1"
name="SecurityGroups">
  <description>Security group support.</description>
</extension>
<extension alias="os-server-diagnostics" updated="2011-12-21T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/server-diagnostics/api/v1.1"
name="ServerDiagnostics">
  <description>Allow Admins to view server diagnostics through server
action.</description>
</extension>
<extension alias="os-server-password" updated="2012-11-29T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/server-password/api/v2"
name="ServerPassword">
  <description>Server password support.</description>
</extension>
<extension alias="os-server-start-stop" updated="2012-01-23T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/servers/api/v1.1" name=
"ServerStartStop">
  <description>Start/Stop instance compute API support.</description>
</extension>
<extension alias="os-services" updated="2012-10-28T00:00:00-00:00"
namespace="http://docs.openstack.org/compute/ext/services/api/v2" name=
"Services">
  <description>Services support.</description>
</extension>
<extension alias="os-shelve" updated="2013-04-06T00:00:00+00:00" namespace=
"http://docs.openstack.org/compute/ext/shelve/api/v1.1" name="Shelve">
  <description>Instance shelve mode.</description>
</extension>
<extension alias="os-simple-tenant-usage" updated=
"2011-08-19T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/
os-simple-tenant-usage/api/v1.1" name="SimpleTenantUsage">
  <description>Simple tenant usage extension.</description>
</extension>
<extension alias="os-tenant-networks" updated="2012-03-07T09:46:43-05:00"
namespace="http://docs.openstack.org/compute/ext/os-tenant-networks/api/v2"
name="OSTenantNetworks">
  <description>Tenant-based Network Management Extension.</description>
</extension>
<extension alias="os-used-limits" updated="2012-07-13T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/used_limits/api/v1.1" name=
"UsedLimits">
  <description>Provide data on limited resources that are being used.</
description>
</extension>
<extension alias="os-used-limits-for-admin" updated=
"2013-05-02T00:00:00+00:00" namespace="http://docs.openstack.org/compute/ext/
used_limits_for_admin/api/v1.1" name="UsedLimitsForAdmin">
  <description>Provide data to admin on limited resources used by other
tenants.</description>

```

```
</extension>
  <extension alias="os-user-data" updated="2012-08-07T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/userdata/api/v1.1" name=
"UserData">
  <description>Add user_data to the Create Server v1.1 API.</description>
</extension>
  <extension alias="os-user-quotas" updated="2013-07-18T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/user_quotas/api/v1.1" name=
"UserQuotas">
  <description>Project user quota support.</description>
</extension>
  <extension alias="os-virtual-interfaces" updated="2011-08-17T00:00:00+00:00"
namespace="http://docs.openstack.org/compute/ext/virtual_interfaces/api/v1.1"
name="VirtualInterfaces">
  <description>Virtual interface support.</description>
</extension>
  <extension alias="os-volume-attachment-update" updated=
"2013-06-20T00:00:00-00:00" namespace="http://docs.openstack.org/compute/ext/
os-volume-attachment-update/api/v2" name="VolumeAttachmentUpdate">
  <description>Support for updating a volume attachment.</description>
</extension>
  <extension alias="os-volumes" updated="2011-03-25T00:00:00+00:00" namespace=
"http://docs.openstack.org/compute/ext/volumes/api/v1.1" name="Volumes">
  <description>Volumes support.</description>
</extension>
</extensions>
```

1.9.2. Get extension

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

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

Normal response codes: 200, 203

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

1.9.2.1. Request

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

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

This operation does not require a request body.

1.9.2.2. Response

Example 1.40. Get extension: JSON response

```
{
  "extension" : {
    "name" : "Public Image Extension",
    "namespace" : "http://docs.rackspacecloud.com/servers/api/ext/pie/v1.0",
    "alias" : "RS-PIE",
    "updated" : "2011-01-22T13:25:27-06:00",
    "description" : "Adds the capability to share an image with other users.",
    "links" : [
      {
        "rel" : "describedby",
        "type" : "application/pdf",
        "href" : "http://docs.rackspacecloud.com/servers/api/ext/cs-pie-20111111.pdf"
      },
      {
        "rel" : "describedby",
        "type" : "application/vnd.sun.wadl+xml",
        "href" : "http://docs.rackspacecloud.com/servers/api/ext/cs-pie.wadl"
      }
    ]
  }
}
```

Example 1.41. Get extension: XML response

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

```
<extension xmlns="http://docs.openstack.org/common/api/v1.0"
  xmlns:atom="http://www.w3.org/2005/Atom"
  name="Public Image Extension"
  namespace="http://docs.rackspacecloud.com/servers/api/ext/pie/v1.0"
  alias="RS-PIE" updated="2011-01-22T13:25:27-06:00">
  <description> Adds the capability to share an image with other
    users. </description>
  <atom:link rel="describedby" type="application/pdf"
    href="http://docs.rackspacecloud.com/servers/api/ext/cs-pie-20111111.
pdf"/>
  <atom:link rel="describedby" type="application/vnd.sun.wadl+xml"
    href="http://docs.rackspacecloud.com/servers/api/ext/cs-pie.wadl"
  />
</extension>
```

1.10. Faults

1.10.1. Synchronous faults

When an error occurs at request time, the system will return an HTTP error response code denoting the type of error. The system will also return additional information about the fault in the body of the response.

Example 1.42. Fault: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<computeFault
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  code="500">
  <message>Fault!</message>
  <details>Error Details...</details>
</computeFault>
```

Example 1.43. Fault: JSON response

```
{
  "computeFault" : {
    "code" : 500,
    "message" : "Fault!",
    "details" : "Error Details..."
  }
}
```

The error code is returned in the body of the response for convenience. The message section returns a human-readable message that is appropriate for display to the end user. The details section is optional and may contain information—for example, a stack trace—to assist in tracking down an error. The detail section may or may not be appropriate for display to an end user.

The root element of the fault (e.g. `computeFault`) may change depending on the type of error. The following is a list of possible elements along with their associated error codes.

Table 1.3. Fault Elements and Error Codes

Fault Element	Associated Error Codes	Expected in All Requests
computeFault	500, 400, other codes possible	✓
notImplemented	501	
serverCapacityUnavailable	503	
serviceUnavailable	503	✓
badRequest	400	✓
unauthorized	401	✓
forbidden	403	✓
resizeNotAllowed	403	
itemNotFound	404	
badMethod	405	
backupOrResizeInProgress	409	
buildInProgress	409	
conflictingRequest	409	
overLimit	413	✓
badMediaType	415	

Example 1.44. Item Not Found fault: JSON response

```
{
  "itemNotFound" : {
    "code" : 404,
    "message" : "Not Found",
    "details" : "Error Details..."
  }
}
```

Example 1.45. Item Not Found fault: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<itemNotFound
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  code="404">
  <message>Not Found</message>
  <details>Error Details...</details>
</itemNotFound>
```

From an XML schema perspective, all API faults are extensions of the base fault type ComputeAPIFault. When working with a system that binds XML to actual classes (such as JAXB), one should be capable of using ComputeAPIFault as a “catch-all” if there’s no interest in distinguishing between individual fault types.

The OverLimit fault is generated when a rate limit threshold is exceeded. For convenience, the fault adds a retryAfter attribute that contains the content of the Retry-After header in XML Schema 1.0 date/time format.

Example 1.46. Over Limit fault: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<overLimit
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  code="413"
  retryAfter="2010-08-01T00:00:00Z">
  <message>OverLimit Retry...</message>
  <details>Error Details...</details>
</overLimit>
```

Example 1.47. Over Limit fault: JSON response

```
{
  "overLimit" : {
    "code" : 413,
    "message" : "OverLimit Retry...",
    "details" : "Error Details...",
    "retryAfter" : "2010-08-01T00:00:00Z"
  }
}
```

1.10.2. Asynchronous faults

An error may occur in the background while a server or image is being built or while a server is executing an action. In these cases, the server or image is placed in an `ERROR` state and the fault is embedded in the offending server or image. Note that these asynchronous faults follow the same format as the synchronous ones. The fault contains an error code, a human readable message, and optional details about the error. Additionally, asynchronous faults may also contain a created timestamp that specify when the fault occurred.

Example 1.48. Server in error state: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  id="52415800-8b69-11e0-9b19-734f0000ffff"
  tenant_id="1234" user_id="5678"
  name="sample-server" status="ERROR"
  created="2010-08-10T12:00:00Z"
  progress="66" hostId="e4d909c290d0fb1ca068ffa0000000000000">
  <image id="52415800-8b69-11e0-9b19-734f6f007777" />
  <flavor id="52415800-8b69-11e0-9b19-734f216543fd" />
  <fault code="404" created="2010-08-10T11:59:59Z">
    <message>Could not find image 52415800-8b69-11e0-9b19-734f6f007777</
message>
    <details>Fault details</details>
  </fault>
  <atom:link
    rel="self"
    href="http://servers.api.openstack.org/v2/1234/servers/
52415800-8b69-11e0-9b19-734f000004d2"/>
  <atom:link
    rel="bookmark"
    href="http://servers.api.openstack.org/1234/servers/
52415800-8b69-11e0-9b19-734f000004d2"/>
</server>
```

Example 1.49. Server in error state: JSON response

```
{
  "server": {
    "id": "52415800-8b69-11e0-9b19-734f0000ffff",
    "tenant_id": "1234",
    "user_id": "5678",
    "name": "sample-server",
    "created": "2010-08-10T12:00:00Z",
    "hostId": "e4d909c290d0fblca068ffa00000000",
    "status": "ERROR",
    "progress": 66,
    "image" : {
      "id": "52415800-8b69-11e0-9b19-734f6f007777"
    },
    "flavor" : {
      "id": "52415800-8b69-11e0-9b19-734f216543fd"
    },
    "fault" : {
      "code" : 404,
      "created": "2010-08-10T11:59:59Z",
      "message": "Could not find image
52415800-8b69-11e0-9b19-734f6f007777",
      "details": "Fault details"
    },
    "links": [
      {
        "rel": "self",
        "href": "http://servers.api.openstack.org/v2/1234/servers/
52415800-8b69-11e0-9b19-734f000004d2"
      },
      {
        "rel": "bookmark",
        "href": "http://servers.api.openstack.org/1234/servers/
52415800-8b69-11e0-9b19-734f000004d2"
      }
    ]
  }
}
```

Example 1.50. Image in error state: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<image
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  id="52415800-8b69-11e0-9b19-734f5736d2a2"
  name="My Server Backup"
  created="2010-08-10T12:00:00Z"
  status="ERROR" progress="89">
  <server id="52415800-8b69-11e0-9b19-734f335aa7b3" />
  <fault code="500">
    <message>An internal error occurred</message>
    <details>Error details</details>
  </fault>
  <atom:link
    rel="self"
    href="http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f5736d2a2"/>
  <atom:link
    rel="bookmark"
    href="http://servers.api.openstack.org/1234/images/
52415800-8b69-11e0-9b19-734f5736d2a2"/>
</image>
```

Example 1.51. Image in error state: JSON response

```
{
  "image" : {
    "id" : "52415800-8b69-11e0-9b19-734f5736d2a2",
    "name" : "My Server Backup",
    "created" : "2010-08-10T12:00:00Z",
    "status" : "SAVING",
    "progress" : 89,
    "server" : {
      "id": "52415800-8b69-11e0-9b19-734f335aa7b3"
    },
    "fault" : {
      "code" : 500,
      "message" : "An internal error occurred",
      "details" : "Error details"
    },
    "links": [
      {
        "rel" : "self",
        "href" : "http://servers.api.openstack.org/v2/1234/images/52415800-8b69-11e0-9b19-734f5736d2a2"
      },
      {
        "rel" : "bookmark",
        "href" : "http://servers.api.openstack.org/1234/images/52415800-8b69-11e0-9b19-734f5736d2a2"
      }
    ]
  }
}
```

2. API operations

2.1. Servers	72
2.2. Server addresses	94
2.3. Server actions	98
2.4. Flavors	109
2.5. Images	114
2.6. Metadata	124
2.7. Networks	142

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

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

2.1. Servers

2.1.1. List servers

You can filter the list of servers by image, flavor, name, and status through the respective query parameters.

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 server status is one of the following values:

Server status values

- **ACTIVE.** The server is active.
- **BUILD.** The server has not finished the original build process.
- **DELETED.** The server is deleted.
- **ERROR.** The server is in error.
- **HARD_REBOOT.** The server is hard rebooting. This is equivalent to pulling the power plug on a physical server, plugging it back in, and rebooting it.
- **PASSWORD.** The password is being reset on the server.
- **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.
- **RESCUE.** The server is in rescue mode.
- **RESIZE.** 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.
- **SHUTOFF.** The virtual machine (VM) was powered down by the user, but not through the OpenStack Compute API. For example, the user issued a `shutdown -h` command from within the server instance. If the OpenStack Compute manager detects that the VM was powered down, it transitions the server instance to the SHUTOFF status. If you use the OpenStack Compute API to restart the instance, the instance might be deleted first, depending on the value in the `shutdown_terminate` database field on the Instance model.

- **SUSPENDED.** The server is suspended, either by request or necessity. This status appears for only the following hypervisors: XenServer/XCP, KVM, and ESXi. Review support tickets or contact Rackspace support to determine why the server is in this state.
- **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.

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. `hostId` represents the host your server runs on and can be used to determine this scenario if it is relevant to your application.



Note

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

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

2.1.1.1. List servers

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

Normal response codes: 200, 203

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

2.1.1.1.1. Request

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

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

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

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

2.1.1.1.2. Response

Example 2.1. List servers: JSON response

```
{
  "servers": [
    {
      "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
```

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

Example 2.2. List servers: XML response

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


2.1.2. Create server

Status Transition:	BUILD → ACTIVE
	BUILD → ERROR (on error)

This operation asynchronously provisions a new server. The progress of this operation depends on several factors including location of the requested image, network i/o, host load, and the selected flavor. The progress of the request can be checked by performing a **GET** on `/servers/id`, which will return a progress attribute (0-100% completion). The full URL to the newly created server is returned via the `Location` header and is available as a `self` and `bookmark` link in the server representation (See [Section 1.4, "Links and references" \[6\]](#)). Note that when creating a server only the server ID, its links, and the admin password are guaranteed to be returned in the request. Additional attributes may be retrieved by performing subsequent **GETs** on the server.

Method	URI	Description
POST	<code>/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}</code>	Creates a server.

2.1.2.1. Create server

Method	URI	Description
POST	/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}	Creates a server.

Normal response codes: 202

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

2.1.2.1.1. Request

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

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

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

Name	Type	Description
security_group	String <i>(Required)</i>	The name of the security group. If blank, the server is created in the default security group.
user_data	String <i>(Optional)</i>	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String <i>(Optional)</i>	The availability zone in which to launch the server.

Example 2.3. Create server: JSON request

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

Example 2.4. Create server: XML request

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

```
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  imageRef="b5660a6e-4b46-4be3-9707-6b47221b454f" flavorRef="2"
  name="server-test-2">
  <metadata>
    <meta key="My Server Name">API Test Server XML</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAGICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
      dCBtb3ZlcyBpbiBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
      IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVz
      c2l2b3I4LnRoXMGaXMGdGhlIHBSYWNlIHRvIGdvIG5vdy4g
      QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
      ZSBwYXR0ZXJucyBiZWwhpbmQgYWxsIGNsb3VkcycwYmV5kIHlv
      dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbG1mdCB5b3Vy
      c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
      b25zLiINCg0KLVJpY2hhcmQgQmFjaA==</file>
    </personality>
  <networks>
    <network uuid="0ef47ac7-6797-4e01-8a47-ed26ec3aaa56"/>
  </networks>
</server>
```

2.1.2.1.2. Response

Example 2.5. Create server: JSON response

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

Example 2.6. Create server: XML response

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

```
xmlns:atom="http://www.w3.org/2005/Atom"
xmlns="http://docs.openstack.org/compute/api/v1.1"
id="9720d338-07fb-4317-9eda-a0a833226afa" adminPass="aabbccddeeff"
OS-DCF:diskConfig="MANUAL">
<metadata/>
<atom:link
  href="http://166.78.46.130:8774/v2/4fd44f30292945e481c7b8a0c8908869/
servers/9720d338-07fb-4317-9eda-a0a833226afa"
  rel="self"/>
<atom:link
  href="http://166.78.46.130:8774/4fd44f30292945e481c7b8a0c8908869/servers/
9720d338-07fb-4317-9eda-a0a833226afa"
  rel="bookmark"/>
<security_groups>
  <security_group name="default"/>
</security_groups>
</server>
```

2.1.2.2. Server passwords

A password may be specified when creating the server via the optional `adminPass` attribute. The specified password must meet the complexity requirements set by your OpenStack Compute provider. The server may enter an `ERROR` state if the complexity requirements are not met. In this case, a client may issue a change password action to reset the server password.

If a password is not specified, a randomly generated password will be assigned and returned in the response object. This password is guaranteed to meet the security requirements set by the compute provider. For security reasons, the password will not be returned in subsequent `GET` calls.

2.1.2.3. Server metadata

Custom server metadata can also be supplied at launch time. See [Section 2.6, "Metadata" \[124\]](#) for details on working with metadata. The maximum size of the metadata key and value is 255 bytes each. The maximum number of key-value pairs that can be supplied per server is determined by the compute provider and may be queried via the `maxServerMeta` absolute limit.

2.1.2.4. Server networks

Networks which the server connects to can also be supplied at launch time. See [Section 2.7, "Networks" \[142\]](#) for details on working with networks. One or more networks can be specified. User can also specify a specific port on the network or the fixed IP address to assign to the server interface.

2.1.2.5. Server personality

You can customize the personality of a server instance by injecting data into its file system. For example, you might want to insert ssh keys, set configuration files, or store data that you want to retrieve from inside the instance. This feature provides a minimal amount of 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 maximum size of the file contents is determined by the compute provider and may vary based on the image that is used to create the server



Note

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

- You can inject text files only. You cannot inject binary or zip files into a new build.
- The maximum number of file path/content pairs that you can supply is also determined by the compute provider and is defined by the `maxPersonality` absolute limit.
- The absolute limit, `maxPersonalitySize`, is a byte limit that is guaranteed to apply to all images in the deployment. Providers can set additional per-image personality limits.

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

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

After file injection, personality files are accessible by only system administrators. For example, on Linux, all files have root and the root group as the owner and group owner, respectively, and allow user and group read access only ().

2.1.2.6. Server access addresses

In a hybrid environment, the IP address of a server may not be controlled by the underlying implementation. Instead, the access IP address may be part of the dedicated hardware; for example, a router/NAT device. In this case, the addresses provided by the implementation cannot actually be used to access the server (from outside the local LAN). Here, a separate *access address* may be assigned at creation time to provide access to the server. This address may not be directly bound to a network interface on the server and may not necessarily appear when a server's addresses are queried. See [Section 2.2, "Server addresses" \[94\]](#). Nonetheless, clients which need to access the server directly are encouraged to do so via an access address. In the example below, an IPv4 address is assigned at creation time.

Example 2.7. Create server with access IP: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  name="test" accessIPv4="67.23.10.132"
  imageRef="52415800-8b69-11e0-9b19-734f6f006e54"
  flavorRef="52415800-8b69-11e0-9b19-734f1195ff37" />
```

Example 2.8. Create server with access IP: JSON request

```
{
  "server" : {
    "name" : "new-server-test",
    "imageRef" : "52415800-8b69-11e0-9b19-734f6f006e54",
    "flavorRef" : "52415800-8b69-11e0-9b19-734f1195ff37",
    "accessIPv4" : "67.23.10.132"
  }
}
```



Note

Both IPv4 and IPv6 addresses may be used as access addresses and both addresses may be assigned simultaneously as illustrated below. Access addresses may be updated after a server has been created. See [Section 2.1, "Servers" \[72\]](#) for more details.

Example 2.9. Create server with multiple access IPs: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  name="test"
  accessIPv4="67.23.10.132"
  accessIPv6="::babe:67.23.10.132"
  imageRef="52415800-8b69-11e0-9b19-734f6f006e54"
  flavorRef="52415800-8b69-11e0-9b19-734f1195ff37" />
```

Example 2.10. Create server with multiple access IPs: JSON request

```
{
  "server" : {
    "name" : "new-server-test",
    "imageRef" : "52415800-8b69-11e0-9b19-734f6f006e54",
    "flavorRef" : "52415800-8b69-11e0-9b19-734f1195ff37",
    "accessIPv4" : "67.23.10.132",
    "accessIPv6" : "::babe:67.23.10.132"
  }
}
```

2.1.3. Get server details

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

2.1.3.1. Get server details

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

Normal response codes: 200, 203

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

2.1.3.1.1. Request

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

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

This operation does not require a request body.

2.1.3.1.2. Response

Example 2.11. Get server details: JSON response

```
{
  "server": {
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "created": "2012-08-20T21:11:09Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "65201c14a29663e06d0748e561207d998b343e1d164bfa0aafa9c45d",
    "id": "893c7791-f1df-4c3d-8383-3caae9656c62",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [

```



```

        "href": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "bookmark"
    }
  ],
  },
  "links": [
    {
      "href": "http://openstack.example.com/v2/openstack/servers/
893c7791-f1df-4c3d-8383-3caae9656c62",
      "rel": "self"
    },
    {
      "href": "http://openstack.example.com/openstack/servers/
893c7791-f1df-4c3d-8383-3caae9656c62",
      "rel": "bookmark"
    }
  ],
  "metadata": {
    "My Server Name": "Apache1"
  },
  "name": "new-server-test",
  "progress": 0,
  "status": "ACTIVE",
  "tenant_id": "openstack",
  "updated": "2012-08-20T21:11:09Z",
  "user_id": "fake"
}
}

```

Example 2.12. Get server details: XML response

```

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

```


2.1.4. Update server

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

2.1.4.1. Update server

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

Normal response codes: 200

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

2.1.4.1.1. Request

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

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

Example 2.13. Update Server Name Request: JSON

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

Example 2.14. Update Server Name Request: XML

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

Example 2.15. Update Server IP Addresses Request: JSON

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

Example 2.16. Update Server IP Addresses Request: XML

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

This operation does not require a request body.

2.1.4.1.2. Response

Example 2.17. Update Server Name Response: JSON

```
{
  "server": {
    "id": "52415800-8b69-11e0-9b19-734f565bc83b",
    "tenant_id": "1234",
    "user_id": "5678",
    "name": "new-server-test",
    "created": "2010-11-11T12:00:00Z",
    "updated": "2010-11-12T12:44:44Z",
    "hostId": "e4d909c290d0fblca068ffaddf22cbd0",
    "accessIPv4" : "67.23.10.138",
    "accessIPv6" : ">::babe:67.23.10.138",
    "progress": 0,
    "status": "ACTIVE",
    "image" : {
      "id": "52415800-8b69-11e0-9b19-734f6f006e54",
      "name": "CentOS 5.2",
      "links": [
        {
          "rel": "self",
          "href": "http://servers.api.openstack.org/v2/1234/images/52415800-8b69-11e0-9b19-734f6f006e54"
        },
        {
          "rel": "bookmark",
          "href": "http://servers.api.openstack.org/1234/images/52415800-8b69-11e0-9b19-734f6f006e54"
        }
      ]
    },
    "flavor" : {
      "id": "52415800-8b69-11e0-9b19-734f1195ff37",
      "name": "256 MB Server",
      "links": [
        {
          "rel": "self",
          "href": "http://servers.api.openstack.org/v2/1234/flavors/52415800-8b69-11e0-9b19-734f1195ff37"
        },
        {
          "rel": "bookmark",
          "href": "http://servers.api.openstack.org/1234/flavors/52415800-8b69-11e0-9b19-734f1195ff37"
        }
      ]
    },
    "metadata": {
      "My Server Name": "Apache1"
    },
    "addresses": {
      "public" : [
        {
          "version": 4,
          "addr": "67.23.10.138"
        },
        {
          "version": 6,
```

```

        "addr": "::babe:67.23.10.138"
      }
    ],
    "private" : [
      {
        "version": 4,
        "addr": "10.176.42.19"
      },
      {
        "version": 6,
        "addr": "::babe:10.176.42.19"
      }
    ]
  },
  "links": [
    {
      "rel": "self",
      "href": "http://servers.api.openstack.org/v2/1234/servers/52415800-8b69-11e0-9b19-734f565bc83b"
    },
    {
      "rel": "bookmark",
      "href": "http://servers.api.openstack.org/1234/servers/52415800-8b69-11e0-9b19-734f565bc83b"
    }
  ]
}

```

Example 2.18. Update Server Name Response: XML

```

<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  id="52415800-8b69-11e0-9b19-734f565bc83b"
  tenant_id="1234" user_id="5678"
  name="new-server-test"
  hostId="e4d909c290d0fb1ca068ffaddf22cbd0" progress="0"
  status="ACTIVE"
  created="2010-11-11T12:00:00Z"
  updated="2010-11-12T12:44:44Z"
  accessIPv4="67.23.10.138"
  accessIPv6 "::babe:67.23.10.138">
  <image id="52415800-8b69-11e0-9b19-734f6f006e54"
    name="CentOS 5.2">
    <atom:link
      rel="self"
      href="http://servers.api.openstack.org/v2/1234/images/52415800-8b69-11e0-9b19-734f6f006e54"/>
    <atom:link
      rel="bookmark"
      href="http://servers.api.openstack.org/1234/images/52415800-8b69-11e0-9b19-734f6f006e54"/>
  </image>
  <flavor id="52415800-8b69-11e0-9b19-734f1195ff37"
    name="256 MB Server">
    <atom:link
      rel="self"
      href="http://servers.api.openstack.org/v2/1234/flavors/52415800-8b69-11e0-9b19-734f1195ff37"/>
  </flavor>
</server>

```

```

    <atom:link
      rel="bookmark"
      href="http://servers.api.openstack.org/1234/flavors/
52415800-8b69-11e0-9b19-734f1195ff37"/>
    </flavor>
  </metadata>
<addresses>
  <network id="public">
    <ip version="4" addr="67.23.10.138"/>
    <ip version="6" addr="::babe:67.23.10.138"/>
  </network>
  <network id="private">
    <ip version="4" addr="10.176.42.19"/>
    <ip version="6" addr="::babe:10.176.42.19"/>
  </network>
</addresses>
<atom:link
  rel="self"
  href="http://servers.api.openstack.org/v2/1234/servers/
52415800-8b69-11e0-9b19-734f565bc83b"/>
<atom:link
  rel="bookmark"
  href="http://servers.api.openstack.org/1234/servers/
52415800-8b69-11e0-9b19-734f565bc83b"/>
</server>

```

Example 2.19. Update Server IP Addresses Response: JSON

```

{
  "server": {
    "id": "52415800-8b69-11e0-9b19-734f565bc83b",
    "tenant_id": "1234",
    "user_id": "5678",
    "name": "new-server-test",
    "created": "2010-11-11T12:00:00Z",
    "updated": "2010-11-12T12:55:55Z",
    "hostId": "e4d909c290d0fb1ca068ffaddf22cbd0",
    "accessIPv4": "67.23.10.132",
    "accessIPv6": "::babe:67.23.10.132",
    "progress": 0,
    "status": "ACTIVE",
    "image": {
      "id": "52415800-8b69-11e0-9b19-734f6f006e54",
      "name": "CentOS 5.2",
      "links": [
        {
          "rel": "self",
          "href": "http://servers.api.openstack.org/v2/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"
        },
        {
          "rel": "bookmark",
          "href": "http://servers.api.openstack.org/1234/images/
52415800-8b69-11e0-9b19-734f6f006e54"
        }
      ]
    },
    "flavor": {

```

```

        "id": "52415800-8b69-11e0-9b19-734f1195ff37",
        "name": "256 MB Server",
        "links": [
            {
                "rel": "self",
                "href": "http://servers.api.openstack.org/v2/1234/flavors/52415800-8b69-11e0-9b19-734f1195ff37"
            },
            {
                "rel": "bookmark",
                "href": "http://servers.api.openstack.org/1234/flavors/52415800-8b69-11e0-9b19-734f1195ff37"
            }
        ]
    },
    "metadata": {
        "My Server Name": "Apache1"
    },
    "addresses": {
        "public" : [
            {
                "version": 4,
                "addr": "67.23.10.138"
            },
            {
                "version": 6,
                "addr": "::babe:67.23.10.138"
            }
        ],
        "private" : [
            {
                "version": 4,
                "addr": "10.176.42.19"
            },
            {
                "version": 6,
                "addr": "::babe:10.176.42.19"
            }
        ]
    },
    "links": [
        {
            "rel": "self",
            "href": "http://servers.api.openstack.org/v2/1234/servers/52415800-8b69-11e0-9b19-734f1195ff37"
        },
        {
            "rel": "bookmark",
            "href": "http://servers.api.openstack.org/1234/servers/52415800-8b69-11e0-9b19-734f1195ff37"
        }
    ]
}

```

Example 2.20. Update Server IP Addresses Response: XML

```

<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
        xmlns:atom="http://www.w3.org/2005/Atom"

```



```

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

```

This operation does not return a response body.

2.1.5. Delete server

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

2.1.5.1. Delete server

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

Normal response codes: 204

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

2.1.5.1.1. Request

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

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

This operation does not require a request body.

2.2. Server addresses

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

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

2.2.1. List addresses

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

Specify the tenant ID and server ID in the URI.

Normal response codes: 200, 203

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

2.2.1.1. Request

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

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

This operation does not require a request body.

2.2.1.2. Response

Example 2.21. List addresses: JSON response

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

```

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

Example 2.22. List addresses: XML response

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

2.2.2. List addresses by network

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

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

Normal response codes: 200, 203

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

2.2.2.1. Request

This table shows the URI parameters for the list addresses by network request:

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

This operation does not require a request body.

2.2.2.2. Response

Example 2.23. List addresses by network: JSON response

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

Example 2.24. List addresses by network: XML response

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

2.3. Server actions

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

2.3.1. Change password

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Changes the password for a server. Specify the <code>changePassword</code> action in the request body.

Normal response codes: 202

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

2.3.1.1. Request

This table shows the URI parameters for the change password request:

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

Example 2.25. Change password: JSON request

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

Example 2.26. Change password: XML request

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


2.3.2. Reboot server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Reboots the specified server. Specify the reboot action in the request body.

Normal response codes: 202

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404), badMediaType (415), HTTPUnprocessableEntity (422), buildInProgress (409)

2.3.2.1. Request

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

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

Example 2.27. Reboot server: JSON request

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

Example 2.28. Reboot server: XML request

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

2.3.3. Rebuild server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Rebuilds the specified server. Specify the rebuild action in the request body.

Normal response codes: 202

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

2.3.3.1. Request

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

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

Example 2.29. Rebuild server: JSON request

```
{
  "rebuild" : {
    "imageRef" : "http://openstack.example.com/v2/32278/images/70a599e0-31e7-49b7-b260-868f441e862b",
    "name" : "foobar",
    "adminPass" : "seekr3t",
    "accessIPv4" : "1.2.3.4",
    "accessIPv6" : "fe80::100",
    "metadata" : {
      "meta var" : "meta val"
    },
    "personality" : [
      {
        "path" : "/etc/banner.txt",
        "contents" : "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcyBpbiBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5kIGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVsc2lvbi4uLnRoXMGaXMgdGhlIHBSYWNlIHRvIGdvIG5vdy4gQnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRoZSBwYXR0ZXJucyBiZWpibmQgYWxsIGNsb3VkcjYwY5kIHlvdSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbG1mdCB5b3VyY2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6b25zLiINCg0KLVJpY2hhcmQgQmFjaA=="
      }
    ]
  }
}
```

Example 2.30. Rebuild server: XML request

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

```

xmlns="http://docs.openstack.org/compute/api/v1.1"
name="foobar"
imageRef="http://openstack.example.com/v1.1/32278/images/70a599e0-31e7-49b7-b260-868f441e862b"
accessIPv4="1.2.3.4"
accessIPv6="fe80::100"
adminPass="seekr3t">
<metadata>
  <meta key="My Server Name">Apache1</meta>
</metadata>
<personality>
  <file path="/etc/banner.txt">
    ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
    dCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
    IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVz
    c2lvbi4uLnRoaXMgaXMgdGhlIHBSYWNlIHRvIGdvIG5vdy4g
    QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
    ZSBwYXR0ZXJucyBjZWwhpbmQgYWxsIGNsb3VkcycwYmV5kIHl
    v
    dSB3aWxsIGtub3csIHRvbywgZ2hlbiB5b3UgbGlmZCB5b3Vy
    c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
    b25zLiINCg0KLVJpY2hhcmQgQmFjaA==
  </file>
</personality>
</rebuild>

```

2.3.3.2. Response

This table shows the header parameters for the rebuild server response:

Name	Type	Description
Location	AnyURI (Required)	Specific URL of the server you want to rebuild.

Example 2.31. Rebuild server: JSON response

```

{
  "server": {
    "accessIPv4": "1.2.3.4",
    "accessIPv6": "fe80::100",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "adminPass": "seekr3t",
    "created": "2012-09-12T17:20:36Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    }
  }
}

```

```

    },
    "hostId": "1e3da81662354c25560b7e5ea6d8123031f67168b6992f20bb84df69",
    "id": "075e40fe-9f03-4652-ba8e-5f8e2547899a",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    },
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/075e40fe-9f03-4652-ba8e-5f8e2547899a",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/075e40fe-9f03-4652-ba8e-5f8e2547899a",
        "rel": "bookmark"
      }
    ],
    "metadata": {
      "meta var": "meta val"
    },
    "name": "foobar",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2012-09-12T17:20:37Z",
    "user_id": "fake"
  }
}

```

Example 2.32. Rebuild server: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" status="ACTIVE" updated="2012-09-14T16:41:46Z" hostId="a0e37e3bd9f674600aabeccb123d80ae2717ace90893d79cd4abc46" name="foobar" created="2012-09-14T16:41:45Z" userId="fake" tenantId="openstack" accessIPv4="1.2.3.4" accessIPv6="fe80::100" progress="0" id="943acea5-2fc8-4f31-bab6-8f7b9ac923ca" adminPass="seekr3t">
  <image id="70a599e0-31e7-49b7-b260-868f441e862b">
    <atom:link href="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  </image>
  <flavor id="1">
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel="bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="private">
      <ip version="4" addr="192.168.0.3"/>
    </network>
  </addresses>
</server>

```

```
</network>  
</addresses>  
  <atom:link href="http://openstack.example.com/v2/openstack/servers/  
943acea5-2fc8-4f31-bab6-8f7b9ac923ca" rel="self"/>  
  <atom:link href="http://openstack.example.com/openstack/servers/  
943acea5-2fc8-4f31-bab6-8f7b9ac923ca" rel="bookmark"/>  
</server>
```

2.3.4. Resize server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Resizes the specified server. Specify the <code>resize</code> action in the request body.

Normal response codes: 202

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

2.3.4.1. Request

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

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

Example 2.33. Resize server: JSON request

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

Example 2.34. Resize server: XML request

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

2.3.5. Confirm resized server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Confirms a pending resize action. Specify the <code>confirmResize</code> action in the request body.

Normal response codes: 204

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

2.3.5.1. Request

This table shows the URI parameters for the confirm resized server request:

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

Example 2.35. Confirm resized server: JSON request

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

Example 2.36. Confirm resized server: XML request

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

2.3.6. Revert resized server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Cancels and reverts a pending resize action. Specify the <code>revertResize</code> action in the request body.

Normal response codes: 202

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

2.3.6.1. Request

This table shows the URI parameters for the revert resized server request:

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

Example 2.37. Revert resized server: JSON request

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

Example 2.38. Revert resized server: XML request

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


2.3.7. Create image

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

Normal response codes: 202

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

2.3.7.1. Request

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

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

Example 2.39. Create image: JSON request

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

Example 2.40. Create image: XML request

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

2.3.7.2. Response

This table shows the header parameters for the create image response:

Name	Type	Description
Location	AnyURI (Required)	

2.4. Flavors

A flavor is an available hardware configuration for a server. Each flavor has a unique combination of disk space and memory capacity.

Method	URI	Description
GET	<code>/v2/flavors/detail{?changes-since, minDisk, minRam, marker, limit}</code>	Lists all details for available flavors.
GET	<code>/v2/flavors/{flavor_id}</code>	Gets details for a specified flavor.

2.4.1. List details for flavors

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

Normal response codes: 200, 203

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

2.4.1.1. Request

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

Name	Type	Description
changes-since	DateTime <i>(Optional)</i>	A time/date stamp for when the flavor last changed.
minDisk	Int <i>(Optional)</i>	Integer value for the minimum disk space in GB so you can filter results.
minRam	Int <i>(Optional)</i>	Integer value for the minimum RAM so you can filter results.
marker	UUID <i>(Optional)</i>	UUID of the flavor at which you want to set a marker.
limit	Int <i>(Optional)</i>	Integer value for the limit of values to return.

2.4.1.2. Response

Example 2.41. List details for flavors: JSON response

```
{
  "flavors": [
    {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/
flavors/1",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/flavors/
1",
          "rel": "bookmark"
        }
      ],
      "name": "m1.tiny"
    },
    {
      "id": "2",
      "links": [
        {
```

```

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

```

```
]
}
```

Example 2.42. List details for flavors: XML response

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

2.4.2. Get flavor details

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

Normal response codes: 200, 203

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

2.4.2.1. Request

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

Name	Type	Description
{flavor_id}	UUID	UUID for the specific flavor (combination of memory, disk size, and CPUs).

This operation does not require a request body.

2.4.2.2. Response

Example 2.43. Get flavor details: JSON response

```
{
  "flavor": {
    "disk": 1,
    "id": "1",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/flavors/1",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/1",
        "rel": "bookmark"
      }
    ],
    "name": "m1.tiny",
    "ram": 512,
    "vcpus": 1
  }
}
```

Example 2.44. Get flavor details: XML response

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

```
</flavor>
```

2.5. Images

An image is a collection of files you use to create or rebuild a server. Operators provide pre-built OS images by default. You may also create custom images.

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

2.5.1. List images details

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

Normal response codes: 200, 203

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

2.5.1.1. Request

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

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

2.5.1.2. Response

Example 2.45. List images details: JSON response

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

```



```

        "rel": "bookmark"
    },
    {
        "href": "http://glance.openstack.example.com/openstack/
images/70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "alternate",
        "type": "application/vnd.openstack.image"
    }
],
"metadata": {
    "architecture": "x86_64",
    "auto_disk_config": "True",
    "kernel_id": "nokernel",
    "ramdisk_id": "nokernel"
},
"minDisk": 0,
"minRam": 0,
"name": "fakeimage7",
"progress": 100,
"status": "ACTIVE",
"updated": "2011-01-01T01:02:03Z"
},
{
    "created": "2011-01-01T01:02:03Z",
    "id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6",
            "rel": "bookmark"
        },
        {
            "href": "http://glance.openstack.example.com/openstack/
images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
            "rel": "alternate",
            "type": "application/vnd.openstack.image"
        }
    ],
    "metadata": {
        "architecture": "x86_64",
        "kernel_id": "nokernel",
        "ramdisk_id": "nokernel"
    },
    "minDisk": 0,
    "minRam": 0,
    "name": "fakeimage123456",
    "progress": 100,
    "status": "ACTIVE",
    "updated": "2011-01-01T01:02:03Z"
},
{
    "created": "2011-01-01T01:02:03Z",
    "id": "a2459075-d96c-40d5-893e-577ff92e721c",
    "links": [
        {

```

```

        "href": "http://openstack.example.com/v2/openstack/images/
a2459075-d96c-40d5-893e-577ff92e721c",
        "rel": "self"
    },
    {
        "href": "http://openstack.example.com/openstack/images/
a2459075-d96c-40d5-893e-577ff92e721c",
        "rel": "bookmark"
    },
    {
        "href": "http://glance.openstack.example.com/openstack/
images/a2459075-d96c-40d5-893e-577ff92e721c",
        "rel": "alternate",
        "type": "application/vnd.openstack.image"
    }
],
"metadata": {
    "kernel_id": "nokernel",
    "ramdisk_id": "nokernel"
},
"minDisk": 0,
"minRam": 0,
"name": "fakeimage123456",
"progress": 100,
"status": "ACTIVE",
"updated": "2011-01-01T01:02:03Z"
},
{
    "created": "2011-01-01T01:02:03Z",
    "id": "a440c04b-79fa-479c-bed1-0b816eaec379",
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/images/
a440c04b-79fa-479c-bed1-0b816eaec379",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/images/
a440c04b-79fa-479c-bed1-0b816eaec379",
            "rel": "bookmark"
        },
        {
            "href": "http://glance.openstack.example.com/openstack/
images/a440c04b-79fa-479c-bed1-0b816eaec379",
            "rel": "alternate",
            "type": "application/vnd.openstack.image"
        }
    ],
    "metadata": {
        "architecture": "x86_64",
        "auto_disk_config": "False",
        "kernel_id": "nokernel",
        "ramdisk_id": "nokernel"
    },
    "minDisk": 0,
    "minRam": 0,
    "name": "fakeimage6",
    "progress": 100,
    "status": "ACTIVE",
    "updated": "2011-01-01T01:02:03Z"
}

```

```

    },
    {
      "created": "2011-01-01T01:02:03Z",
      "id": "c905cedb-7281-47e4-8a62-f26bc5fc4c77",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77",
          "rel": "bookmark"
        },
        {
          "href": "http://glance.openstack.example.com/openstack/
images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
          "rel": "alternate",
          "type": "application/vnd.openstack.image"
        }
      ],
      "metadata": {
        "kernel_id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
        "ramdisk_id": null
      },
      "minDisk": 0,
      "minRam": 0,
      "name": "fakeimage123456",
      "progress": 100,
      "status": "ACTIVE",
      "updated": "2011-01-01T01:02:03Z"
    },
    {
      "created": "2011-01-01T01:02:03Z",
      "id": "cedef40a-ed67-4d10-800e-17455edce175",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175",
          "rel": "bookmark"
        },
        {
          "href": "http://glance.openstack.example.com/openstack/
images/cedef40a-ed67-4d10-800e-17455edce175",
          "rel": "alternate",
          "type": "application/vnd.openstack.image"
        }
      ],
      "metadata": {
        "kernel_id": "nokernel",
        "ramdisk_id": "nokernel"
      },
      "minDisk": 0,
      "minRam": 0,

```

```

        "name": "fakeimage123456",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    },
    {
        "created": "2011-01-01T01:02:03Z",
        "id": "76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
                "rel": "alternate",
                "type": "application/vnd.openstack.image"
            }
        ],
        "metadata": {
            "kernel_id": "nokernel",
            "ramdisk_id": "nokernel"
        },
        "minDisk": 0,
        "minRam": 0,
        "name": "fakeimage123456",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    }
]
}

```

Example 2.46. List images details: XML response

```

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

```

```

    <atom:link href="http://glance.openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"155d900f-4e14-4e4c-a73d-069cbf4541e6">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="ramdisk_id">nokernel</meta>
      <meta key="architecture">x86_64</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"a2459075-d96c-40d5-893e-577ff92e721c">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="ramdisk_id">nokernel</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
a2459075-d96c-40d5-893e-577ff92e721c" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/a2459075-
d96c-40d5-893e-577ff92e721c" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
a2459075-d96c-40d5-893e-577ff92e721c" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage6"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"a440c04b-79fa-479c-bed1-0b816eaec379">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="auto_disk_config">False</meta>
      <meta key="ramdisk_id">nokernel</meta>
      <meta key="architecture">x86_64</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
a440c04b-79fa-479c-bed1-0b816eaec379" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/
a440c04b-79fa-479c-bed1-0b816eaec379" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
a440c04b-79fa-479c-bed1-0b816eaec379" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"c905cedb-7281-47e4-8a62-f26bc5fc4c77">
    <metadata>
      <meta key="kernel_id">155d900f-4e14-4e4c-a73d-069cbf4541e6</meta>
      <meta key="ramdisk_id">None</meta>
    </metadata>

```

```

    <atom:link href="http://openstack.example.com/v2/openstack/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"cedef40a-ed67-4d10-800e-17455edce175">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="ramdisk_id">nokernel</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/cedef40a-
ed67-4d10-800e-17455edce175" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"76fa36fc-c930-4bf3-8c8a-ea2a2420deb6">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="ramdisk_id">nokernel</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/76fa36fc-
c930-4bf3-8c8a-ea2a2420deb6" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
</images>

```

2.5.2. Get image details

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

Normal response codes: 200, 203

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

2.5.2.1. Request

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

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

This operation does not require a request body.

2.5.2.2. Response

Example 2.47. Get image details: JSON response

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

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

Example 2.48. Get image details: XML response

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


2.5.3. Delete image

Method	URI	Description
DELETE	/v2/images/{image_id}	Deletes a specified image.

Normal response codes: 204

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

2.5.3.1. Request

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

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

This operation does not require a request body.

2.6. Metadata

The following operations enable access to metadata after a server or image has been created.

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

2.6.1. Show server metadata

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

Normal response codes: 200, 203

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

2.6.1.1. Request

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

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

This operation does not require a request body.

2.6.1.2. Response

Example 2.49. Show server metadata: JSON response

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

Example 2.50. Show server metadata: XML response

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

2.6.2. Create or replace server metadata

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

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

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

Normal response codes: 200

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

2.6.2.1. Request

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

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

Example 2.51. Create or replace server metadata: JSON request

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

Example 2.52. Create or replace server metadata: XML request

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

2.6.2.2. Response

Example 2.53. Create or replace server metadata: JSON response

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

Example 2.54. Create or replace server metadata: XML response

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

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

2.6.3. Update server metadata items

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

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

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

Normal response codes: 200

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

2.6.3.1. Request

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

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

Example 2.55. Update server metadata items: JSON request

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

Example 2.56. Update server metadata items: XML request

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

2.6.3.2. Response

Example 2.57. Update server metadata items: JSON response

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

Example 2.58. Update server metadata items: XML response

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

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

2.6.4. Show server metadata item details

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

Normal response codes: 200, 203

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

2.6.4.1. Request

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

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

This operation does not require a request body.

2.6.4.2. Response

Example 2.59. Show server metadata item details: JSON response

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

Example 2.60. Show server metadata item details: XML response

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

2.6.5. Create or update server metadata item

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

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

Normal response codes: 200

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

2.6.5.1. Request

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

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

Example 2.61. Create or update server metadata item: JSON request

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

Example 2.62. Create or update server metadata item: XML request

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

2.6.5.2. Response

Example 2.63. Create or update server metadata item: JSON response

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

Example 2.64. Create or update server metadata item: XML response

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



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

2.6.6. Delete server metadata item

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

Normal response codes: 204

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

2.6.6.1. Request

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

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

This operation does not require a request body.

2.6.7. Show image metadata

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

Normal response codes: 200, 203

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

2.6.7.1. Request

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

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

This operation does not require a request body.

2.6.7.2. Response

Example 2.65. Show image metadata: JSON response

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

Example 2.66. Show image metadata: XML response

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

2.6.8. Create or replace image metadata

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

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

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

Normal response codes: 200

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

2.6.8.1. Request

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

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

Example 2.67. Create or replace image metadata: JSON request

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

Example 2.68. Create or replace image metadata: XML request

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

2.6.8.2. Response

Example 2.69. Create or replace image metadata: JSON response

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

Example 2.70. Create or replace image metadata: XML response

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

2.6.9. Update image metadata items

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

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

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

Normal response codes: 200

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

2.6.9.1. Request

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

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

Example 2.71. Update image metadata items: JSON request

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

Example 2.72. Update image metadata items: XML request

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

2.6.9.2. Response

Example 2.73. Update image metadata items: JSON response

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

Example 2.74. Update image metadata items: XML response

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

2.6.10. Show image metadata item details

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

Normal response codes: 200, 203

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

2.6.10.1. Request

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

Name	Type	Description
{image_id}	UUID	The UUID for the image.
{key}	Image Metadata Key	A string. Maximum length is 255 characters.

This operation does not require a request body.

2.6.10.2. Response

Example 2.75. Show image metadata item details: JSON response

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

Example 2.76. Show image metadata item details: XML response

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


2.6.11. Create or update image metadata item

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

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

Normal response codes: 200

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

2.6.11.1. Request

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

Name	Type	Description
{image_id}	UUID	The UUID for the image.
{key}	Image Metadata Key	A string. Maximum length is 255 characters.

Example 2.77. Create or update image metadata item: JSON request

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

Example 2.78. Create or update image metadata item: XML request

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

2.6.11.2. Response

Example 2.79. Create or update image metadata item: JSON response

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

Example 2.80. Create or update image metadata item: XML response

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

2.6.12. Delete image metadata item

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

Normal response codes: 204

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

2.6.12.1. Request

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

Name	Type	Description
{image_id}	UUID	The UUID for the image.
{key}	Image Metadata Key	A string. Maximum length is 255 characters.

This operation does not require a request body.

2.7. Networks

You can specify the following attributes for a network.

Table 2.1. Network attributes

Name	Type	Description
uuid	uuid	The uuid of the network.
fixed_ip	IPv4	The IP address to assign to the interface.
port	uuid	The uuid of the port.



Note

The *fixed_ip* parameter is used only when network *uuid* is specified; also, when *port* is specified, network *uuid* and *fixed_ip* are properties of the port and are ignored. Omit *fixed_ip* and (network) *uuid* to avoid validation errors.

3. Compute API v2 extensions

3.1. Server admin actions (action)	152
3.2. Server console output (os-console-output)	169
3.3. Server console (os-consoles)	170
3.4. Server deferred delete (os-deferred-delete)	173
3.5. Server diagnostics (diagnostics)	175
3.6. Flavor access (flavors)	176
3.7. Flavors with FlavorDisabled attribute (flavors)	189
3.8. Flavor extra-specs (os-extra-specs)	194
3.9. Flavors with rxtx_factor extended attribute (flavors)	200
3.10. Flavors with extended attributes (flavors)	207
3.11. Flavors create or delete (flavors)	215
3.12. Images with size attribute (images)	219
3.13. Limits with project usage (limits)	228
3.14. Limits with project usage for administrators (limits)	231
3.15. Guest agents (os-agents)	235
3.16. Host aggregates (os-aggregates)	242
3.17. Attach interfaces (os-attach-interfaces)	257
3.18. Root certificates (os-certificates)	263
3.19. Cloudpipe (os-cloudpipe)	268
3.20. Coverage reports (os-coverage)	271
3.21. Fixed IPs (os-fixed-ips)	275
3.22. Floating IP DNS records (os-floating-ip-dns)	278
3.23. Floating IP pools (os-floating-ip-pools)	287
3.24. Floating IPs (os-floating-ips)	288
3.25. Floating IPs bulk (os-floating-ips-bulk)	296
3.26. Hosts (os-hosts)	302
3.27. Hypervisors (os-hypervisors)	311
3.28. Server actions (os-instance-actions)	318
3.29. Keypairs (os-keypairs)	323
3.30. Migrations (os-migrations)	328
3.31. Networks (os-networks)	330
3.32. Quota sets (os-quota-sets)	343
3.33. Server rescue and unrescue (os-rescue)	355
3.34. Rules for default security group (os-security-group-default-rules)	357
3.35. Security groups (os-security-groups)	361
3.36. Server password (os-server-password)	368
3.37. Server shelve (os-server-shelve)	371
3.38. Server start and stop (os-server-start-stop)	372
3.39. Manage services (os-services)	374
3.40. Usage reports (os-simple-tenant-usage)	381
3.41. Virtual interfaces (os-virtual-interfaces)	384
3.42. Volume extension (os-volumes, os-snapshots)	386
3.43. Volume attachments (os-volume_attachments)	405
3.44. Servers with block device mapping format (servers)	411
3.45. Server OS-EXT-IPS-MAC:mac_addr extended attribute (servers)	416
3.46. Configuration drive (servers)	425
3.47. Servers with extended availability zones (servers)	434
3.48. Servers and images with disk config (servers, images)	439

3.49. Server IP type (servers) 462
 3.50. Server extended attributes (servers) 467
 3.51. Server extended status (servers) 473
 3.52. Servers multiple create (servers) 479
 3.53. Servers with scheduler hints (servers) 483

Extensions add features, MIME types, actions, states, headers, parameters, and resources to the core Compute API. Query the Compute API to list available extensions with a **GET** request to `v2/extensions`.

Method	URI	Description
Server admin actions (action)		
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Pauses a server. Changes its status to PAUSED.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Unpauses a PAUSED server and changes its status to ACTIVE.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Suspends a server and changes its status to SUSPENDED.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Resumes a SUSPENDED server and changes its status to ACTIVE.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Migrates a server to a host. The scheduler chooses the host.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Resets networking on a server.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Injects network information into a server.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Locks a server.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Unlocks a server.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Backs up a server instance.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Live-migrates a server to a new host without rebooting.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Resets the state of a server to a specified state.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Evacuates a server from failed host.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Assigns the specified security group to the server.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Removes the specified security group from the server.
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Adds a floating IP address to an instance.
Server console output (os-console-output)		
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Gets console output for a server instance.
Server console (os-consoles)		
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Gets a console for a server instance.
Server deferred delete (os-deferred-delete)		
POST	<code>/v2/{tenant_id}/servers/{server_id}/action</code>	Force-deletes a server.

Method	URI	Description
POST	/v2/{tenant_id}/servers/ {server_id}/action	Restores a deleted server.
Server diagnostics (diagnostics)		
GET	/v2/{tenant_id}/servers/ {server_id}/diagnostics	Gets basic usage data for a specified server.
Flavor access (flavors)		
GET	/v2/{tenant_id}/flavors	Lists flavors and includes the access type, which is public or private.
POST	/v2/{tenant_id}/flavors	Creates a private flavor.
GET	/v2/{tenant_id}/flavors/ {flavor_id}	Gets the flavor access type, which is public or private.
GET	/v2/{tenant_id}/flavors/ {flavor_id}/os-flavor-access	Lists tenants with access to the specified private flavor.
POST	/v2/{tenant_id}/flavors/ {flavor_id}/action	Gives a specified tenant access to the specified private flavor.
DELETE	/v2/{tenant_id}/flavors/ {flavor_id}/action	Revokes access from the specified tenant for the specified private flavor.
Flavors with FlavorDisabled attribute (flavors)		
GET	/v2/{tenant_id}/flavors/ {flavor_id}	Gets details for a specified flavor. Includes the OS-FLV-DISABLED:disabled attribute.
GET	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the OS-FLV-DISABLED:disabled attribute.
Flavor extra-specs (os-extra-specs)		
GET	/v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs	Lists the extra-specs or keys for the specified flavor.
POST	/v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs	Creates extra-specs or keys for the specified flavor.
GET	/v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs/ {key_id}	Gets the value of the specified key.
DELETE	/v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs/ {key_id}	Deletes a specified extra-spec by key.
Flavors with rxtx_factor extended attribute (flavors)		
POST	/v2/{tenant_id}/flavors	Creates a flavor. Includes the rxtx_factor extended attribute.
GET	/v2/{tenant_id}/flavors/ {flavor_id}	Gets details for a specified flavor. Includes the rxtx_factor extended attribute.
GET	/v2/{tenant_id}/flavors/detail	Lists details for available flavors and includes the rxtx_factor extended attribute.
Flavors with extended attributes (flavors)		
POST	/v2/{tenant_id}/flavors	Creates a flavor. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.
GET	/v2/{tenant_id}/flavors/ {flavor_id}	Gets details for a specified flavor. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.
GET	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.
Flavors create or delete (flavors)		
POST	/v2/{tenant_id}/flavors	Creates a flavor.
DELETE	/v2/{tenant_id}/flavors/ {flavor_id}	Deletes a flavor.

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

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

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

Method	URI	Description
POST	/v2/{tenant_id}/os-networks/{id}/action	Disassociates the project from a specified network.
Quota sets (os-quota-sets)		
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Shows quotas for a tenant.
PUT	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.
GET	/v2/{tenant_id}/os-quota-sets/defaults	Gets default quotas for a tenant.
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a specified tenant and user.
POST	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a specified tenant/project and user.
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}	Shows details for quotas for a specified tenant and user.
Server rescue and unrescue (os-rescue)		
POST	/v2/{tenant_id}/servers/{server_id}/action	Puts a server in rescue mode. Changes status to RESCUE.
POST	/v2/{tenant_id}/servers/{server_id}/action	Unrescues a server.
Rules for default security group (os-security-group-default-rules)		
GET	/v2/{tenant_id}/os-security-group-rules	Lists default security group rules.
POST	/v2/{tenant_id}/os-security-group-rules	Creates a default security group rule.
GET	/v2/{tenant_id}/os-security-group-rules/{security_group_rule_id}	Shows information for a specified security group rule.
Security groups (os-security-groups)		
GET	/v2/{tenant_id}/os-security-groups	Lists security groups.
POST	/v2/{tenant_id}/os-security-groups	Creates a security group.
GET	/v2/{tenant_id}/os-security-groups/servers/{server_id}/os-security-groups	Lists security groups for a specified server.
GET	/v2/{tenant_id}/os-security-groups/{security_group_id}	Shows information for a specified security group.
DELETE	/v2/{tenant_id}/os-security-groups/{security_group_id}	Deletes a specified security group.
Server password (os-server-password)		
GET	/v2/servers/{server_id}/os-server-password	Gets the administrative password for a specified server.
DELETE	/v2/servers/{server_id}/os-server-password	Clears the encrypted copy of the password in the metadata server. This is done after the client has retrieved the password and knows it doesn't need it in the metadata server anymore. The password for the server remains the same.
Server shelve (os-server-shelve)		
POST	/v2/{tenant_id}/servers/{server_id}/action	Shelves a running server and changes its status to SHELVED_OFFLOADED.
Server start and stop (os-server-start-stop)		
POST	/v2/{tenant_id}/servers/{server_id}/action	Starts a stopped server and changes its status to ACTIVE.

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Stops a running server and changes its status to STOPPED.
Manage services (os-services)		
GET	/v2/{tenant_id}/os-services	Lists running services.
PUT	/v2/{tenant_id}/os-services/enable	Enables scheduling for a service.
PUT	/v2/{tenant_id}/os-services/disable	Disables scheduling for a service.
PUT	/v2/{tenant_id}/os-services/disable-log-reason	Logs information to the service table about why a service was disabled.
GET	/v2/{tenant_id}/os-services/detail	Lists disabled services. If information exists, includes reasons why services were disabled.
Usage reports (os-simple-tenant-usage)		
GET	/v2/{tenant_id}/os-simple-tenant-usage	Lists usage information for all tenants.
GET	/v2/{tenant_id}/os-simple-tenant-usage/{tenant_id}	Gets usage information for a tenant.
Virtual interfaces (os-virtual-interfaces)		
GET	/v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Lists the virtual interfaces for a specified instance.
GET	/v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Shows the virtual interface for a specified instance. Includes the OS-EXT-VIF-NET:net_id attribute that shows to which network the interface is attached.
Volume extension (os-volumes, os-snapshots)		
GET	/v1.1/{tenant_id}/os-volumes	Lists the volumes associated with the account.
GET	/v1.1/{tenant_id}/os-volumes/detail	Lists details for a specified volume.
POST	/v1.1/{tenant_id}/os-volumes/{volume_id}	Creates a volume.
GET	/v1.1/{tenant_id}/os-volumes/{volume_id}	Shows information for a specified volume.
DELETE	/v1.1/{tenant_id}/os-volumes/{volume_id}	Deletes a specified volume.
GET	/v1.1/{tenant_id}/os-volume-types	Lists volume types.
GET	/v1.1/{tenant_id}/os-volume-types/{volume_type_id}	Shows information for a specified volume type.
POST	/v1.1/{tenant_id}/os-snapshots	Creates a snapshot.
GET	/v1.1/{tenant_id}/os-snapshots	Lists snapshots.
GET	/v1.1/{tenant_id}/os-snapshots/detail	Lists details for a specified snapshot.
GET	/v1.1/{tenant_id}/os-snapshots/{snapshot_id}	Shows information for a specified snapshot.
DELETE	/v1.1/{tenant_id}/os-snapshots/{snapshot_id}	Deletes a specified snapshot from the account.
Volume attachments (os-volume_attachments)		
POST	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Attaches a volume to the specified server.
GET	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Lists the volume attachments for a specified server.
GET	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Shows details for the specified volume attachment.

Method	URI	Description
DELETE	/v2/{tenant_id}/servers/{server_id}/os-volume-attachments/{attachment_id}	Deletes the specified volume attachment from a specified server.
Servers with block device mapping format (servers)		
GET	/v2/{tenant_id}/servers{?changesince,image,flavor,name,marker,limit,status,host}	Lists IDs, names, and links for all servers.
POST	/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}	Creates a server with a block device mapping.
Server OS-EXT-IPS-MAC:mac_addr extended attribute (servers)		
POST	/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}	Creates a server with the OS-EXT-IPS-MAC:mac_addr extended attribute.
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.
GET	/v2/{tenant_id}/servers/detail	Lists details for all servers. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.
Configuration drive (servers)		
POST	/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}	Creates a server with the configuration drive extended attribute.
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server including the configuration drive extended attribute.
GET	/v2/{tenant_id}/servers/{server_id}/detail	Lists details for all servers including the configuration drive extended attribute.
Servers with extended availability zones (servers)		
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server, including its availability zone.
GET	/v2/{tenant_id}/servers/detail	Lists details for servers, including their current availability zone.
Servers and images with disk config (servers, images)		
POST	/v2/{tenant_id}/servers	Creates a server.
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server.
PUT	/v2/{tenant_id}/servers/{server_id}	Updates a specified server.
POST	/v2/{tenant_id}/servers/{server_id}/action	Resizes a server.
POST	/v2/{tenant_id}/servers/{server_id}/action	Rebuilds a specified server.
GET	/v2/{tenant_id}/servers/detail	Lists servers.
GET	/v2/{tenant_id}/images/{image_id}	Gets information for a specified image.
GET	/v2/{tenant_id}/images/detail	Lists images.
Server IP type (servers)		
GET	/v2/{tenant_id}/servers/{server_id}/action	Shows the type of IP assigned to a specified server, either fixed or floating.
GET	/v2/{tenant_id}/servers/detail	Lists all servers showing their IPs by type, either fixed or floating.
Server extended attributes (servers)		
GET	/v2/servers	Lists detailed extended server attribute information for all servers.

Method	URI	Description
GET	/v2/servers/{server_id}	Shows extended server attributes for a specified server.
Server extended status (servers)		
GET	/v2/{tenant_id}/servers/{server_id}	Shows the extended status attributes in the response for a specified server.
GET	/v2/{tenant_id}/servers/detail	Lists the extended status attributes in the detailed response for all servers.
Servers multiple create (servers)		
POST	/v2/{tenant_id}/servers{? security_group,user_data, availability_zone, return_reservation_id,min_count, max_count}	Creates one or more servers with an optional reservation ID.
Servers with scheduler hints (servers)		
POST	/v2/{tenant_id}/servers{? security_group,user_data, availability_zone}	Creates a server with scheduler hints that are passed directly to the scheduler.

3.1. Server admin actions (action)

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

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Pauses a server. Changes its status to PAUSED.
POST	/v2/{tenant_id}/servers/{server_id}/action	Unpauses a PAUSED server and changes its status to ACTIVE.
POST	/v2/{tenant_id}/servers/{server_id}/action	Suspends a server and changes its status to SUSPENDED.
POST	/v2/{tenant_id}/servers/{server_id}/action	Resumes a SUSPENDED server and changes its status to ACTIVE.
POST	/v2/{tenant_id}/servers/{server_id}/action	Migrates a server to a host. The scheduler chooses the host.
POST	/v2/{tenant_id}/servers/{server_id}/action	Resets networking on a server.
POST	/v2/{tenant_id}/servers/{server_id}/action	Injects network information into a server.
POST	/v2/{tenant_id}/servers/{server_id}/action	Locks a server.
POST	/v2/{tenant_id}/servers/{server_id}/action	Unlocks a server.
POST	/v2/{tenant_id}/servers/{server_id}/action	Backs up a server instance.
POST	/v2/{tenant_id}/servers/{server_id}/action	Live-migrates a server to a new host without rebooting.
POST	/v2/{tenant_id}/servers/{server_id}/action	Resets the state of a server to a specified state.
POST	/v2/{tenant_id}/servers/{server_id}/action	Evacuates a server from failed host.
POST	/v2/{tenant_id}/servers/{server_id}/action	Assigns the specified security group to the server.
POST	/v2/{tenant_id}/servers/{server_id}/action	Removes the specified security group from the server.

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

3.1.1. Pause server

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

Normal response codes: 202

3.1.1.1. Request

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

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

Example 3.1. Pause server: JSON request

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

Example 3.2. Pause server: XML request

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

This operation does not require a request body.

3.1.2. Unpause server

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

Normal response codes: 202

3.1.2.1. Request

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

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

Example 3.3. Unpause server: JSON request

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

Example 3.4. Unpause server: XML request

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

This operation does not require a request body.

3.1.3. Suspend server

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

Normal response codes: 202

3.1.3.1. Request

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

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

Example 3.5. Suspend server: JSON request

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

Example 3.6. Suspend server: XML request

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

This operation does not require a request body.

3.1.4. Resume server

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

Normal response codes: 202

3.1.4.1. Request

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

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

Example 3.7. Resume server: JSON request

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

Example 3.8. Resume server: XML request

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

This operation does not require a request body.

3.1.5. Migrate server

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

Normal response codes: 202

3.1.5.1. Request

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

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

Example 3.9. Migrate server: JSON request

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

Example 3.10. Migrate server: xml request

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

This operation does not require a request body.

3.1.6. Reset networking on server

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

Normal response codes: 202

3.1.6.1. Request

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

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

Example 3.11. Reset network: JSON request

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

Example 3.12. Reset network: XML request

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

This operation does not require a request body.

3.1.7. Inject network information

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

Normal response codes: 202

3.1.7.1. Request

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

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

Example 3.13. Insert network information: JSON request

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

Example 3.14. Insert network information: XML request

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

This operation does not require a request body.

3.1.8. Lock server

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

Normal response codes: 202

3.1.8.1. Request

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

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

Example 3.15. Lock server: JSON request

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

Example 3.16. Lock server: XML request

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

This operation does not require a request body.

3.1.9. Unlock server

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

Normal response codes: 202

3.1.9.1. Request

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

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

Example 3.17. Unlock server: JSON request

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

Example 3.18. Unlock server: XML request

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

This operation does not require a request body.

3.1.10. Create server backup

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

Normal response codes: 202

3.1.10.1. Request

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

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

Example 3.19. Create server backup: JSON request

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

Example 3.20. Create server backup: XML request

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

This operation does not require a request body.

3.1.11. Live-migrate server

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

Normal response codes: 202

3.1.11.1. Request

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

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

Example 3.21. Live-migrate server: JSON request

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

Example 3.22. Live-migrate server: XML request

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

This operation does not require a request body.

3.1.12. Reset server state

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

Normal response codes: 202

3.1.12.1. Request

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

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

Example 3.23. Reset server state: JSON request

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

Example 3.24. Reset server state: XML request

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

This operation does not require a request body.

3.1.13. Evacuate server

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

Normal response codes: 200

3.1.13.1. Request

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

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

Example 3.25. Evacuate server: JSON request

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

Example 3.26. Evacuate server: XML request

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

This operation does not require a request body.

3.1.13.2. Response

Example 3.27. Evacuate server: JSON response

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

Example 3.28. Evacuate server: XML response

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

This operation does not return a response body.

3.1.14. Add security group

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

Normal response codes: 200

3.1.14.1. Request

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

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

Example 3.29. Add security group: JSON request

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

Example 3.30. Add security group: XML request

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

This operation does not require a request body.

3.1.15. Remove security group

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

Normal response codes: 200

3.1.15.1. Request

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

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

Example 3.31. Remove security group: JSON request

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

Example 3.32. Remove security group: XML request

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

This operation does not require a request body.

3.1.16. Add floating IP address

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

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

Normal response codes: 200

3.1.16.1. Request

This table shows the URI parameters for the add floating ip address request:

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

Example 3.33. Add floating IP address: JSON request

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

3.2. Server console output (os-console-output)

Get console output for a server instance.

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

3.2.1. Get console output for an instance

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

Normal response codes: 200

3.2.1.1. Request

This table shows the URI parameters for the get console output for an instance request:

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

Example 3.34. Get console output: JSON request

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

Example 3.35. Get console output: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<os-getConsoleOutput length="50" />
```

This operation does not require a request body.

3.2.1.2. Response

Example 3.36. Get console output: JSON response

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

Example 3.37. Get console output: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<output>FAKE CONSOLE OUTPUT
ANOTHER
LAST LINE</output>
```

This operation does not return a response body.

3.3. Server console (os-consoles)

Get a console for a server instance.

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

3.3.1. Get console

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

Normal response codes: 200

3.3.1.1. Request

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

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

Example 3.38. Get console: JSON request

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

Example 3.39. Get console: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<os-getVNCConsole type="novnc" />
```

This operation does not require a request body.

3.3.1.2. Response

Example 3.40. Get console: JSON response

```
{
  "console": {
    "type": "novnc",
    "url": "http://example.com:6080/vnc_auto.html?token=f9906a48-b71e-4f18-baca-c987da3ebdb3&title=dafa(75ecef58-3b8e-4659-ab3b-5501454188e9)"
  }
}
```

Example 3.41. Get console: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<console>
  <type>novnc</type>
  <url>http://example.com:6080/vnc_auto.html?token=f9906a48-b71e-4f18-baca-c987da3ebdb3</url>
</console>
```

This operation does not return a response body.

3.4. Server deferred delete (os-deferred-delete)

Force-delete a server or restore a deleted server.

Method	URI	Description
POST	/v2/{tenant_id}/servers/ {server_id}/action	Force-deletes a server.
POST	/v2/{tenant_id}/servers/ {server_id}/action	Restores a deleted server.

3.4.1. Force delete server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Force-deletes a server.

Normal response codes: 202

3.4.1.1. Request

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

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

Example 3.42. Force delete server: JSON request

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

Example 3.43. Force delete server: XML request

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

This operation does not require a request body.

3.4.2. Restore server

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

Normal response codes: 202

3.4.2.1. Request

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

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

Example 3.44. Restore server: JSON request

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

Example 3.45. Restore server: XML request

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

This operation does not require a request body.

3.5. Server diagnostics (diagnostics)

Get the usage data for a server.

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

3.5.1. Get server diagnostics

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

Normal response codes: 200

3.5.1.1. Request

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

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

This operation does not require a request body.

3.5.1.2. Response

Example 3.46. Server diagnostics: JSON response

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

3.6. Flavor access (flavors)

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

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

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

3.6.1. List flavors with access type

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

Normal response codes: 200

3.6.1.1. Request

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

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

This operation does not require a request body.

3.6.1.2. Response

Example 3.47. List flavors with access type: JSON response

```
{
  "flavors": [
    {
      "disk": 1,
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/
flavors/1",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/flavors/
1",
          "rel": "bookmark"
        }
      ],
      "name": "m1.tiny",
      "os-flavor-access:is_public": true,
      "ram": 512,
      "vcpus": 1
    },
    {
      "disk": 20,
      "id": "2",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/
flavors/2",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/flavors/
2",
          "rel": "bookmark"
        }
      ]
    }
  ]
}
```

```

    ],
    "name": "m1.small",
    "os-flavor-access:is_public": true,
    "ram": 2048,
    "vcpus": 1
  },
  {
    "disk": 40,
    "id": "3",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/3",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
3",
        "rel": "bookmark"
      }
    ],
    "name": "m1.medium",
    "os-flavor-access:is_public": true,
    "ram": 4096,
    "vcpus": 2
  },
  {
    "disk": 80,
    "id": "4",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/4",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
4",
        "rel": "bookmark"
      }
    ],
    "name": "m1.large",
    "os-flavor-access:is_public": true,
    "ram": 8192,
    "vcpus": 4
  },
  {
    "disk": 160,
    "id": "5",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/5",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
5",
        "rel": "bookmark"
      }
    ]
  }

```



```
    ],
    "name": "m1.xlarge",
    "os-flavor-access:is_public": true,
    "ram": 16384,
    "vcpus": 8
  }
]
```

Example 3.48. List flavors with access type: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:os-flavor-access="http://docs.openstack.org/compute/ext/
flavor_access/api/v2" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/compute/api/v1.1">
  <flavor disk="1" vcpus="1" ram="512" name="m1.tiny" id="1" os-flavor-
access:is_public="True">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
  </flavor>
  <flavor disk="20" vcpus="1" ram="2048" name="m1.small" id="2" os-flavor-
access:is_public="True">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/2" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/2" rel=
"bookmark"/>
  </flavor>
  <flavor disk="40" vcpus="2" ram="4096" name="m1.medium" id="3" os-flavor-
access:is_public="True">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/3" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/3" rel=
"bookmark"/>
  </flavor>
  <flavor disk="80" vcpus="4" ram="8192" name="m1.large" id="4" os-flavor-
access:is_public="True">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/4" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/4" rel=
"bookmark"/>
  </flavor>
  <flavor disk="160" vcpus="8" ram="16384" name="m1.xlarge" id="5" os-flavor-
access:is_public="True">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/5" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/5" rel=
"bookmark"/>
  </flavor>
</flavors>
```

This operation does not return a response body.

3.6.2. Create private flavor

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

Normal response codes: 200

3.6.2.1. Request

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

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

Example 3.49. Create private flavor: JSON request

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

Example 3.50. Create private flavor: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:os-flavor-access="http://docs.openstack.org/compute/ext/
flavor_access/api/v1.1"
  name="test_flavor"
  ram="1024"
  vcpus="2"
  disk="10"
  id="10"
  os-flavor-access:is_public="False"
/>
```

This operation does not require a request body.

3.6.2.2. Response

Example 3.51. Create private flavor: JSON response

```
{
  "flavor": {
    "disk": 10,
    "id": "10",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/flavors/
10",
```

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

Example 3.52. Create private flavor: XML response

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

This operation does not return a response body.

3.6.3. Show flavor access type

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

Normal response codes: 200

3.6.3.1. Request

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

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

This operation does not require a request body.

3.6.3.2. Response

Example 3.53. Show flavor access type: JSON response

```
{
  "flavor": {
    "disk": 1,
    "id": "1",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/flavors/1",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/1",
        "rel": "bookmark"
      }
    ],
    "name": "m1.tiny",
    "os-flavor-access:is_public": true,
    "ram": 512,
    "vcpus": 1
  }
}
```

Example 3.54. Show flavor access type: XML response

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

This operation does not return a response body.

3.6.4. List tenants with access to private flavor

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

Normal response codes: 200

3.6.4.1. Request

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

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

This operation does not require a request body.

3.6.4.2. Response

Example 3.55. List tenants with access to private flavor: JSON response

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

Example 3.56. List tenants with access to private flavor: XML response

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

This operation does not return a response body.

3.6.5. Add access to private flavor

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

Normal response codes: 200

3.6.5.1. Request

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

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

Example 3.57. Add access to private flavor: JSON request

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

Example 3.58. Add access to private flavor: XML request

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

This operation does not require a request body.

3.6.5.2. Response

Example 3.59. Add access to private flavor: JSON response

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

Example 3.60. Add access to private flavor: XML response

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

```
<access tenant_id="fake_tenant" flavor_id="10"/>  
<access tenant_id="openstack" flavor_id="10"/>  
</flavor_access>
```

This operation does not return a response body.

3.6.6. Delete access from private flavor

Method	URI	Description
DELETE	/v2/{tenant_id}/flavors/{flavor_id}/action	Revokes access from the specified tenant for the specified private flavor.

Normal response codes: 200

3.6.6.1. Request

This table shows the URI parameters for the delete access from private flavor request:

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

Example 3.61. Delete access from private flavor: JSON request

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

Example 3.62. Delete access from private flavor: XML request

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

This operation does not require a request body.

3.6.6.2. Response

Example 3.63. Delete access from private flavor: JSON response

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

Example 3.64. Delete access from private flavor: XML response

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

This operation does not return a response body.

3.7. Flavors with FlavorDisabled attribute (flavors)

Get details for a flavor, and list details for available flavors. Includes the OS-FLV-DISABLED:disabled extended attribute.

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the OS-FLV-DISABLED:disabled attribute.
GET	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the OS-FLV-DISABLED:disabled attribute.

3.7.1. Get flavor disabled status details

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the OS-FLV-DISABLED:disabled attribute.

Normal response codes: 200200

3.7.1.1. Request

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

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

This operation does not require a request body.

3.7.1.2. Response

Example 3.65. Get flavor disabled status details: JSON response

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

Example 3.66. Get flavor disabled status details: XML response

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

This operation does not return a response body.

3.7.2. List flavors with flavor disabled status

Method	URI	Description
GET	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the OS-FLV-DISABLED:disabled attribute.

Normal response codes: 200200

3.7.2.1. Request

This table shows the URI parameters for the list flavors with flavor disabled status request:

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

This operation does not require a request body.

3.7.2.2. Response

Example 3.67. List flavors with flavor disabled status: JSON response

```
{
  "flavors": [
    {
      "OS-FLV-DISABLED:disabled": false,
      "disk": 1,
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/
flavors/1",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/flavors/
1",
          "rel": "bookmark"
        }
      ],
      "name": "m1.tiny",
      "ram": 512,
      "vcpus": 1
    },
    {
      "OS-FLV-DISABLED:disabled": false,
      "disk": 20,
      "id": "2",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/
flavors/2",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/flavors/
2",
          "rel": "bookmark"
        }
      ]
    }
  ]
}
```

```

        }
    ],
    "name": "m1.small",
    "ram": 2048,
    "vcpus": 1
  },
  {
    "OS-FLV-DISABLED:disabled": false,
    "disk": 40,
    "id": "3",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/3",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
3",
        "rel": "bookmark"
      }
    ],
    "name": "m1.medium",
    "ram": 4096,
    "vcpus": 2
  },
  {
    "OS-FLV-DISABLED:disabled": false,
    "disk": 80,
    "id": "4",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/4",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
4",
        "rel": "bookmark"
      }
    ],
    "name": "m1.large",
    "ram": 8192,
    "vcpus": 4
  },
  {
    "OS-FLV-DISABLED:disabled": false,
    "disk": 160,
    "id": "5",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/5",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
5",
        "rel": "bookmark"
      }
    ]
  }
]

```

```
    }  
    ],  
    "name": "m1.xlarge",  
    "ram": 16384,  
    "vcpus": 8  
  }  
]  
}
```

Example 3.68. List flavors with flavor disabled status: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<flavors xmlns:OS-FLV-DISABLED="http://docs.openstack.org/compute/ext/  
flavor_disabled/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=  
"http://docs.openstack.org/compute/api/v1.1">  
  <flavor disk="1" vcpus="1" ram="512" name="m1.tiny" id="1" OS-FLV-  
DISABLED:disabled="False">  
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=  
"self"/>  
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=  
"bookmark"/>  
  </flavor>  
  <flavor disk="20" vcpus="1" ram="2048" name="m1.small" id="2" OS-FLV-  
DISABLED:disabled="False">  
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/2" rel=  
"self"/>  
    <atom:link href="http://openstack.example.com/openstack/flavors/2" rel=  
"bookmark"/>  
  </flavor>  
  <flavor disk="40" vcpus="2" ram="4096" name="m1.medium" id="3" OS-FLV-  
DISABLED:disabled="False">  
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/3" rel=  
"self"/>  
    <atom:link href="http://openstack.example.com/openstack/flavors/3" rel=  
"bookmark"/>  
  </flavor>  
  <flavor disk="80" vcpus="4" ram="8192" name="m1.large" id="4" OS-FLV-  
DISABLED:disabled="False">  
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/4" rel=  
"self"/>  
    <atom:link href="http://openstack.example.com/openstack/flavors/4" rel=  
"bookmark"/>  
  </flavor>  
  <flavor disk="160" vcpus="8" ram="16384" name="m1.xlarge" id="5" OS-FLV-  
DISABLED:disabled="False">  
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/5" rel=  
"self"/>  
    <atom:link href="http://openstack.example.com/openstack/flavors/5" rel=  
"bookmark"/>  
  </flavor>  
</flavors>
```

This operation does not return a response body.

3.8. Flavor extra-specs (os-extra-specs)

List, create, and update the extra-specs or keys for a flavor.

Method	URI	Description
GET	/v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs	Lists the extra-specs or keys for the specified flavor.
POST	/v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs	Creates extra-specs or keys for the specified flavor.
GET	/v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs/ {key_id}	Gets the value of the specified key.
DELETE	/v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs/ {key_id}	Deletes a specified extra-spec by key.

3.8.1. List flavor extra specs

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs	Lists the extra-specs or keys for the specified flavor.

Normal response codes: 200

3.8.1.1. Request

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

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

This operation does not require a request body.

3.8.1.2. Response

Example 3.69. List flavor extra specs: JSON response

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

Example 3.70. List flavor extra specs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extra_specs>
  <key2>value2</key2>
  <key1>value1</key1>
</extra_specs>
```

This operation does not return a response body.

3.8.2. Create flavor extra specs

Method	URI	Description
POST	/v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs	Creates extra-specs or keys for the specified flavor.

Normal response codes: 200

3.8.2.1. Request

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

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

Example 3.71. Create flavor extra specs: JSON request

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

Example 3.72. Create flavor extra specs: XML request

```
<?xml version="1.0" encoding="UTF-8" ?>
<extra_specs>
  <key1>value1</key1>
  <key2>value2</key2>
</extra_specs>
```

This operation does not require a request body.

3.8.2.2. Response

Example 3.73. Create flavor extra specs: JSON response

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

Example 3.74. Create flavor extra specs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extra_specs>
  <key2>value2</key2>
  <key1>value1</key1>
</extra_specs>
```

This operation does not return a response body.

3.8.3. Get flavor extra spec details

Method	URI	Description
GET	/v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs/ {key_id}	Gets the value of the specified key.

Normal response codes: 200

3.8.3.1. Request

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

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

This operation does not require a request body.

3.8.3.2. Response

Example 3.75. Get flavor extra spec details: JSON response

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

Example 3.76. Get flavor extra spec details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extra_spec key="key1">value1</extra_spec>
```

This operation does not return a response body.

3.8.4. Delete flavor extra specs

Method	URI	Description
DELETE	/v2/{tenant_id}/flavors/ {flavor_id}/os-extra_specs/ {key_id}	Deletes a specified extra-spec by key.

Normal response codes: 200

3.8.4.1. Request

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

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

This operation does not require a request body.

3.9. Flavors with rxtx_factor extended attribute (flavors)

Create a flavor, get details for a specified flavor, and list details for available flavors. Includes the `rxtx_factor` extended attribute, related to configured bandwidth cap values.

Method	URI	Description
POST	/v2/{tenant_id}/flavors	Creates a flavor. Includes the <code>rxtx_factor</code> extended attribute.
GET	/v2/{tenant_id}/flavors/ {flavor_id}	Gets details for a specified flavor. Includes the <code>rxtx_factor</code> extended attribute.
GET	/v2/{tenant_id}/flavors/detail	Lists details for available flavors and includes the <code>rxtx_factor</code> extended attribute.

3.9.1. Create flavor with rxtx_factor

Method	URI	Description
POST	/v2/{tenant_id}/flavors	Creates a flavor. Includes the rxtx_factor extended attribute.

Normal response codes: 200

3.9.1.1. Request

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

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

Example 3.77. Create flavor with rxtx_factor: JSON request

```
{
  "flavor": {
    "name": "flavortest",
    "ram": 1024,
    "vcpus": 2,
    "disk": 10,
    "id": "100",
    "rxtx_factor": 2.0
  }
}
```

Example 3.78. Create flavor with rxtx_factor: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<flavor xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:OS-FLV-EXT-DATA="http://docs.openstack.org/compute/ext/
flavor_extra_data/api/v1.1"
  name="flavortest"
  ram="1024"
  vcpus="2"
  disk="10"
  id="100"
  rxtx_factor="2.0" />
```

This operation does not require a request body.

3.9.1.2. Response

Example 3.79. Create flavor with rxtx_factor: JSON response

```
{
  "flavor": {
    "disk": 10,
    "id": "100",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/flavors/
100",
```

```
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/100",
        "rel": "bookmark"
      }
    ],
    "name": "flavortest",
    "ram": 1024,
    "rxtx_factor": 2.0,
    "vcpus": 2
  }
}
```

Example 3.80. Create flavor with `rxtx_factor`: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" disk="10" vcpus="2" ram="1024" name="flavortest" id="100" rxtx_factor="2.0">
  <atom:link href="http://openstack.example.com/v2/openstack/flavors/100" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/flavors/100" rel="bookmark"/>
</flavor>
```

This operation does not return a response body.

3.9.2. Get flavor with rxtx_factor

Method	URI	Description
GET	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the rxtx_factor extended attribute.

Normal response codes: 200200

3.9.2.1. Request

This table shows the URI parameters for the get flavor with rxtx_factor request:

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

This operation does not require a request body.

3.9.2.2. Response

Example 3.81. Get flavor with rxtx_factor: JSON response

```
{
  "flavor": {
    "disk": 1,
    "id": "1",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/flavors/1",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/1",
        "rel": "bookmark"
      }
    ],
    "name": "m1.tiny",
    "ram": 512,
    "rxtx_factor": 1.0,
    "vcpus": 1
  }
}
```

Example 3.82. Get flavor with rxtx_factor: XML response

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


This operation does not return a response body.

3.9.3. Get flavor Details with rxtx_factor

Method	URI	Description
GET	/v2/{tenant_id}/flavors/detail	Lists details for available flavors and includes the rxtx_factor extended attribute.

Normal response codes: 200

3.9.3.1. Request

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

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

This operation does not require a request body.

3.9.3.2. Response

Example 3.83. Get flavor Details with rxtx_factor: JSON response

```
{
  "flavors": [
    {
      "disk": 1,
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/
flavors/1",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/flavors/
1",
          "rel": "bookmark"
        }
      ],
      "name": "m1.tiny",
      "ram": 512,
      "rxtx_factor": 1.0,
      "vcpus": 1
    },
    {
      "disk": 20,
      "id": "2",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/
flavors/2",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/flavors/
2",
          "rel": "bookmark"
        }
      ]
    }
  ]
}
```

```

    ],
    "name": "m1.small",
    "ram": 2048,
    "rxtx_factor": 1.0,
    "vcpus": 1
  },
  {
    "disk": 40,
    "id": "3",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/3",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
3",
        "rel": "bookmark"
      }
    ],
    "name": "m1.medium",
    "ram": 4096,
    "rxtx_factor": 1.0,
    "vcpus": 2
  },
  {
    "disk": 80,
    "id": "4",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/4",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
4",
        "rel": "bookmark"
      }
    ],
    "name": "m1.large",
    "ram": 8192,
    "rxtx_factor": 1.0,
    "vcpus": 4
  },
  {
    "disk": 160,
    "id": "5",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/5",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
5",
        "rel": "bookmark"
      }
    ]
  }

```

```
    ],  
    "name": "m1.xlarge",  
    "ram": 16384,  
    "rxtx_factor": 1.0,  
    "vcpus": 8  
  }  
]  
}
```

Example 3.84. Get flavor Details with rxtx_factor: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<flavors xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.  
openstack.org/compute/api/v1.1">  
  <flavor disk="1" vcpus="1" ram="512" name="m1.tiny" id="1" rxtx_factor="1.  
0">  
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=  
"self"/>  
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=  
"bookmark"/>  
  </flavor>  
  <flavor disk="20" vcpus="1" ram="2048" name="m1.small" id="2" rxtx_factor=  
"1.0">  
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/2" rel=  
"self"/>  
    <atom:link href="http://openstack.example.com/openstack/flavors/2" rel=  
"bookmark"/>  
  </flavor>  
  <flavor disk="40" vcpus="2" ram="4096" name="m1.medium" id="3" rxtx_factor=  
"1.0">  
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/3" rel=  
"self"/>  
    <atom:link href="http://openstack.example.com/openstack/flavors/3" rel=  
"bookmark"/>  
  </flavor>  
  <flavor disk="80" vcpus="4" ram="8192" name="m1.large" id="4" rxtx_factor=  
"1.0">  
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/4" rel=  
"self"/>  
    <atom:link href="http://openstack.example.com/openstack/flavors/4" rel=  
"bookmark"/>  
  </flavor>  
  <flavor disk="160" vcpus="8" ram="16384" name="m1.xlarge" id="5"  
rxtx_factor="1.0">  
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/5" rel=  
"self"/>  
    <atom:link href="http://openstack.example.com/openstack/flavors/5" rel=  
"bookmark"/>  
  </flavor>  
</flavors>
```

This operation does not return a response body.

3.10. Flavors with extended attributes (flavors)

Create a flavor, get details for a flavor, and list details for available flavors. Includes the `rxtx_factor`, `OS-FLV-EXT-DATA:ephemeral`, and `swap` extended attributes.

Method	URI	Description
POST	/v2/{tenant_id}/flavors	Creates a flavor. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.
GET	/v2/{tenant_id}/flavors/{flavor_id}	Gets details for a specified flavor. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.
GET	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.

3.10.1. Create flavor with extra data

Method	URI	Description
POST	/v2/{tenant_id}/flavors	Creates a flavor. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.

Normal response codes: 200

3.10.1.1. Request

This table shows the URI parameters for the create flavor with extra data request:

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

Example 3.85. Create flavor with extra data: JSON request

```
{
  "flavor": {
    "name": "flavortest",
    "ram": 1024,
    "vcpus": 2,
    "disk": 10,
    "id": "666",
    "rxtx_factor": 2.0,
    "OS-FLV-EXT-DATA:ephemeral": 30,
    "swap": 5
  }
}
```

Example 3.86. Create flavor with extra data: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<flavor xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:OS-FLV-EXT-DATA="http://docs.openstack.org/compute/ext/
flavor_extra_data/api/v1.1"
  name="flavortest"
  ram="1024"
  vcpus="2"
  disk="10"
  id="666"
  swap="5"
  rxtx_factor="2.0"
  OS-FLV-EXT-DATA:ephemeral="30" />
```

This operation does not require a request body.

3.10.1.2. Response

Example 3.87. Create flavor with extra data: JSON response

```
{
  "flavor": {
    "OS-FLV-EXT-DATA:ephemeral": 30,
    "disk": 10,
```

```
        "id": "666",
        "links": [
          {
            "href": "http://openstack.example.com/v2/openstack/flavors/
666",
            "rel": "self"
          },
          {
            "href": "http://openstack.example.com/openstack/flavors/666",
            "rel": "bookmark"
          }
        ],
        "name": "flavortest",
        "ram": 1024,
        "vcpus": 2
      }
    ]
  }
```

Example 3.88. Create flavor with extra data: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:OS-FLV-EXT-DATA="http://docs.openstack.org/compute/ext/
flavor_extra_data/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1" disk="10" vcpus="2" ram="1024"
name="flavortest" id="666" OS-FLV-EXT-DATA:ephemeral="30">
  <atom:link href="http://openstack.example.com/v2/openstack/flavors/666" rel=
"self"/>
  <atom:link href="http://openstack.example.com/openstack/flavors/666" rel=
"bookmark"/>
</flavor>
```

This operation does not return a response body.

3.10.2. Get flavor extra data details

Method	URI	Description
GET	/v2/{tenant_id}/flavors/ {flavor_id}	Gets details for a specified flavor. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.

Normal response codes: 200200

3.10.2.1. Request

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

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

This operation does not require a request body.

3.10.2.2. Response

Example 3.89. Get flavor extra data details: JSON response

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

Example 3.90. Get flavor extra data details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavor xmlns:OS-FLV-EXT-DATA="http://docs.openstack.org/compute/ext/
flavor_extra_data/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1" disk="1" vcpus="1" ram="512"
name="m1.tiny" id="1" OS-FLV-EXT-DATA:ephemeral="0">
  <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
  <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
</flavor>
```



```
</flavor>
```

This operation does not return a response body.

3.10.3. List flavors with extra data

Method	URI	Description
GET	/v2/{tenant_id}/flavors/detail	Lists available flavors. Includes the rxtx_factor, OS-FLV-EXT-DATA:ephemeral, and swap extended attributes.

Normal response codes: 200200

3.10.3.1. Request

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

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

This operation does not require a request body.

3.10.3.2. Response

Example 3.91. List flavors with extra data: JSON response

```
{
  "flavors": [
    {
      "OS-FLV-EXT-DATA:ephemeral": 0,
      "disk": 1,
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/
flavors/1",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/flavors/
1",
          "rel": "bookmark"
        }
      ],
      "name": "m1.tiny",
      "ram": 512,
      "vcpus": 1
    },
    {
      "OS-FLV-EXT-DATA:ephemeral": 0,
      "disk": 20,
      "id": "2",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/
flavors/2",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/flavors/
2",
          "rel": "bookmark"
        }
      ]
    }
  ]
}
```

```

    }
    ],
    "name": "m1.small",
    "ram": 2048,
    "vcpus": 1
  },
  {
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "disk": 40,
    "id": "3",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/3",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
3",
        "rel": "bookmark"
      }
    ],
    "name": "m1.medium",
    "ram": 4096,
    "vcpus": 2
  },
  {
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "disk": 80,
    "id": "4",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/4",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
4",
        "rel": "bookmark"
      }
    ],
    "name": "m1.large",
    "ram": 8192,
    "vcpus": 4
  },
  {
    "OS-FLV-EXT-DATA:ephemeral": 0,
    "disk": 160,
    "id": "5",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/
flavors/5",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/
5",
        "rel": "bookmark"
      }
    ]
  }
]

```

```
    }
    ],
    "name": "m1.xlarge",
    "ram": 16384,
    "vcpus": 8
  }
]
}
```

Example 3.92. List flavors with extra data: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<flavors xmlns:OS-FLV-EXT-DATA="http://docs.openstack.org/compute/ext/
flavor_extra_data/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1">
  <flavor disk="1" vcpus="1" ram="512" name="m1.tiny" id="1" OS-FLV-EXT-
DATA:ephemeral="0">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/1" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
  </flavor>
  <flavor disk="20" vcpus="1" ram="2048" name="m1.small" id="2" OS-FLV-EXT-
DATA:ephemeral="0">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/2" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/2" rel=
"bookmark"/>
  </flavor>
  <flavor disk="40" vcpus="2" ram="4096" name="m1.medium" id="3" OS-FLV-EXT-
DATA:ephemeral="0">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/3" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/3" rel=
"bookmark"/>
  </flavor>
  <flavor disk="80" vcpus="4" ram="8192" name="m1.large" id="4" OS-FLV-EXT-
DATA:ephemeral="0">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/4" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/4" rel=
"bookmark"/>
  </flavor>
  <flavor disk="160" vcpus="8" ram="16384" name="m1.xlarge" id="5" OS-FLV-EXT-
DATA:ephemeral="0">
    <atom:link href="http://openstack.example.com/v2/openstack/flavors/5" rel=
"self"/>
    <atom:link href="http://openstack.example.com/openstack/flavors/5" rel=
"bookmark"/>
  </flavor>
</flavors>
```

This operation does not return a response body.

3.11. Flavors create or delete (flavors)

Create or delete flavors.

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

3.11.1. Create flavor

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

Normal response codes: 200

3.11.1.1. Request

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

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

Example 3.93. Create flavor: JSON request

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

Example 3.94. Create flavor: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<flavor>
  <name>test_flavor</name>
  <ram>1024</ram>
  <vcpus>2</vcpus>
  <disk>10</disk>
  <id>10</id>
</flavor>
```

This operation does not require a request body.

3.11.1.2. Response

Example 3.95. Create flavor: JSON response

```
{
  "flavor": {
    "disk": 10,
    "id": "10",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/flavors/10",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/flavors/10",
```

```
        "rel": "bookmark"
      }
    ],
    "name": "test_flavor",
    "ram": 1024,
    "vcpus": 2
  }
}
```

Example 3.96. Create flavor: XML response

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

This operation does not return a response body.

3.11.2. Delete flavor

Method	URI	Description
DELETE	/v2/{tenant_id}/flavors/ {flavor_id}	Deletes a flavor.

Normal response codes: 204

3.11.2.1. Request

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

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

This operation does not require a request body.

3.12. Images with size attribute (images)

List details for available images or get details for a specified image. Includes the `OS-EXT-IMG-SIZE:size` extended attribute, which shows the image size.

Method	URI	Description
GET	/v2/{tenant_id}/images/detail	Lists details for available images. Includes the image size.
GET	/v2/{tenant_id}/images/{image_id}	Gets details for a specified image. Includes the image size.

3.12.1. List details for images

Method	URI	Description
GET	/v2/{tenant_id}/images/detail	Lists details for available images. Includes the image size.

Normal response codes: 200200

3.12.1.1. Request

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

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

This operation does not require a request body.

3.12.1.2. Response

Example 3.97. List details for images: JSON response

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

```

    {
      "OS-EXT-IMG-SIZE:size": "25165824",
      "created": "2011-01-01T01:02:03Z",
      "id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
          "rel": "bookmark"
        },
        {
          "href": "http://glance.openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
          "rel": "alternate",
          "type": "application/vnd.openstack.image"
        }
      ],
      "metadata": {
        "architecture": "x86_64",
        "kernel_id": "nokernel",
        "ramdisk_id": "nokernel"
      },
      "minDisk": 0,
      "minRam": 0,
      "name": "fakeimage123456",
      "progress": 100,
      "status": "ACTIVE",
      "updated": "2011-01-01T01:02:03Z"
    },
    {
      "OS-EXT-IMG-SIZE:size": "58145823",
      "created": "2011-01-01T01:02:03Z",
      "id": "a2459075-d96c-40d5-893e-577ff92e721c",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",
          "rel": "bookmark"
        },
        {
          "href": "http://glance.openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",
          "rel": "alternate",
          "type": "application/vnd.openstack.image"
        }
      ],
      "metadata": {
        "kernel_id": "nokernel",
        "ramdisk_id": "nokernel"
      },
    }
  ]
}

```

```

        "minDisk": 0,
        "minRam": 0,
        "name": "fakeimage123456",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    },
    {
        "OS-EXT-IMG-SIZE:size": "49163826",
        "created": "2011-01-01T01:02:03Z",
        "id": "a440c04b-79fa-479c-bed1-0b816eae379",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/images/a440c04b-79fa-479c-bed1-0b816eae379",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eae379",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eae379",
                "rel": "alternate",
                "type": "application/vnd.openstack.image"
            }
        ],
        "metadata": {
            "architecture": "x86_64",
            "auto_disk_config": "False",
            "kernel_id": "nokernel",
            "ramdisk_id": "nokernel"
        },
        "minDisk": 0,
        "minRam": 0,
        "name": "fakeimage6",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    },
    {
        "OS-EXT-IMG-SIZE:size": "26360814",
        "created": "2011-01-01T01:02:03Z",
        "id": "c905cedb-7281-47e4-8a62-f26bc5fc4c77",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",

```

```

        "rel": "alternate",
        "type": "application/vnd.openstack.image"
    },
    ],
    "metadata": {
        "kernel_id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
        "ramdisk_id": null
    },
    "minDisk": 0,
    "minRam": 0,
    "name": "fakeimage123456",
    "progress": 100,
    "status": "ACTIVE",
    "updated": "2011-01-01T01:02:03Z"
},
{
    "OS-EXT-IMG-SIZE:size": "84035174",
    "created": "2011-01-01T01:02:03Z",
    "id": "cedef40a-ed67-4d10-800e-17455edce175",
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175",
            "rel": "bookmark"
        },
        {
            "href": "http://glance.openstack.example.com/openstack/
images/cedef40a-ed67-4d10-800e-17455edce175",
            "rel": "alternate",
            "type": "application/vnd.openstack.image"
        }
    ],
    "metadata": {
        "kernel_id": "nokernel",
        "ramdisk_id": "nokernel"
    },
    "minDisk": 0,
    "minRam": 0,
    "name": "fakeimage123456",
    "progress": 100,
    "status": "ACTIVE",
    "updated": "2011-01-01T01:02:03Z"
},
{
    "OS-EXT-IMG-SIZE:size": "83594576",
    "created": "2011-01-01T01:02:03Z",
    "id": "76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
            "rel": "self"
        },
    ],
}

```

```

        "href": "http://openstack.example.com/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
        "rel": "bookmark"
    },
    {
        "href": "http://glance.openstack.example.com/openstack/
images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
        "rel": "alternate",
        "type": "application/vnd.openstack.image"
    }
],
"metadata": {
    "kernel_id": "nokernel",
    "ramdisk_id": "nokernel"
},
"minDisk": 0,
"minRam": 0,
"name": "fakeimage123456",
"progress": 100,
"status": "ACTIVE",
"updated": "2011-01-01T01:02:03Z"
}
]
}

```

Example 3.98. List details for images: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<images xmlns:OS-EXT-IMG-SIZE="http://docs.openstack.org/compute/ext/
image_size/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/compute/api/v1.1">
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage7"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"70a599e0-31e7-49b7-b260-868f441e862b" OS-EXT-IMG-SIZE:size="74185822">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="auto_disk_config">True</meta>
      <meta key="ramdisk_id">nokernel</meta>
      <meta key="architecture">x86_64</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"155d900f-4e14-4e4c-a73d-069cbf4541e6" OS-EXT-IMG-SIZE:size="25165824">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="ramdisk_id">nokernel</meta>
      <meta key="architecture">x86_64</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="bookmark"/>
  </image>
</images>

```

```

    <atom:link href="http://glance.openstack.example.com/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"a2459075-d96c-40d5-893e-577ff92e721c" OS-EXT-IMG-SIZE:size="58145823">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="ramdisk_id">nokernel</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
a2459075-d96c-40d5-893e-577ff92e721c" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/a2459075-
d96c-40d5-893e-577ff92e721c" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
a2459075-d96c-40d5-893e-577ff92e721c" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage6"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"a440c04b-79fa-479c-bed1-0b816eaec379" OS-EXT-IMG-SIZE:size="49163826">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="auto_disk_config">False</meta>
      <meta key="ramdisk_id">nokernel</meta>
      <meta key="architecture">x86_64</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
a440c04b-79fa-479c-bed1-0b816eaec379" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/
a440c04b-79fa-479c-bed1-0b816eaec379" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
a440c04b-79fa-479c-bed1-0b816eaec379" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"c905cedb-7281-47e4-8a62-f26bc5fc4c77" OS-EXT-IMG-SIZE:size="26360814">
    <metadata>
      <meta key="kernel_id">155d900f-4e14-4e4c-a73d-069cbf4541e6</meta>
      <meta key="ramdisk_id">None</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"cedef40a-ed67-4d10-800e-17455edce175" OS-EXT-IMG-SIZE:size="84035174">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="ramdisk_id">nokernel</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175" rel="self"/>

```

```
<atom:link href="http://openstack.example.com/openstack/images/cedef40a-
ed67-4d10-800e-17455edce175" rel="bookmark"/>
  <atom:link href="http://glance.openstack.example.com/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175" type="application/vnd.openstack.image"
rel="alternate"/>
</image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" OS-EXT-IMG-SIZE:size="83594576">
  <metadata>
    <meta key="kernel_id">nokernel</meta>
    <meta key="ramdisk_id">nokernel</meta>
  </metadata>
  <atom:link href="http://openstack.example.com/v2/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/images/76fa36fc-
c930-4bf3-8c8a-ea2a2420deb6" rel="bookmark"/>
  <atom:link href="http://glance.openstack.example.com/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" type="application/vnd.openstack.image"
rel="alternate"/>
</image>
</images>
```

This operation does not return a response body.

3.12.2. Get image details

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

Normal response codes: 200200

3.12.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{image_id}	String	Image ID stored through the image API. Typically a UUID.

This operation does not require a request body.

3.12.2.2. Response

Example 3.99. Get image details: JSON response

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



```

    "updated": "2011-01-01T01:02:03Z"
  }
}

```

Example 3.100. Get image details: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<image xmlns:OS-EXT-IMG-SIZE="http://docs.openstack.org/compute/ext/
image_size/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1" status="ACTIVE" updated=
"2011-01-01T01:02:03Z" name="fakeimage7" created="2011-01-01T01:02:03Z"
minDisk="0" progress="100" minRam="0" id="70a599e0-31e7-49b7-
b260-868f441e862b" OS-EXT-IMG-SIZE:size="74185822">
  <metadata>
    <meta key="kernel_id">nokernel</meta>
    <meta key="auto_disk_config">True</meta>
    <meta key="ramdisk_id">nokernel</meta>
    <meta key="architecture">x86_64</meta>
  </metadata>
  <atom:link href="http://openstack.example.com/v2/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  <atom:link href="http://glance.openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" type="application/vnd.openstack.image"
rel="alternate"/>
</image>

```

This operation does not return a response body.

3.13. Limits with project usage (limits)

Extend limits to show the project usage. Show information such as RAM or instance quotas usage.

Method	URI	Description
GET	/v2/{tenant_id}/limits	Gets absolute and rate limit information, including information on currently used absolute limits.

3.13.1. Get limits

Method	URI	Description
GET	/v2/{tenant_id}/limits	Gets absolute and rate limit information, including information on currently used absolute limits.

Normal response codes: 200

3.13.1.1. Request

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

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

This operation does not require a request body.

3.13.1.2. Response

Example 3.101. Get limits: JSON response

```
{
  "limits": {
    "absolute": {
      "maxImageMeta": 128,
      "maxPersonality": 5,
      "maxPersonalitySize": 10240,
      "maxSecurityGroupRules": 20,
      "maxSecurityGroups": 10,
      "maxServerMeta": 128,
      "maxTotalCores": 20,
      "maxTotalFloatingIps": 10,
      "maxTotalInstances": 10,
      "maxTotalKeypairs": 100,
      "maxTotalRAMSize": 51200,
      "totalCoresUsed": 0,
      "totalInstancesUsed": 0,
      "totalRAMUsed": 0,
      "totalSecurityGroupsUsed": 0,
      "totalFloatingIpsUsed": 0
    },
    "rate": [
      {
        "limit": [
          {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "POST"
          }
        ]
      },
      {
        "next-available": "2012-11-27T17:24:52Z",
        "remaining": 120,
        "unit": "MINUTE",
        "value": 120,
        "verb": "POST"
      }
    ]
  }
}
```

```

        "value": 120,
        "verb": "PUT"
    },
    {
        "next-available": "2012-11-27T17:24:52Z",
        "remaining": 120,
        "unit": "MINUTE",
        "value": 120,
        "verb": "DELETE"
    }
],
"regex": ".*",
"uri": "*"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "POST"
        }
    ],
    "regex": "^/servers",
    "uri": "*/servers"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "GET"
        }
    ],
    "regex": ".*changes-since.*",
    "uri": "*changes-since*"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 12,
            "unit": "MINUTE",
            "value": 12,
            "verb": "GET"
        }
    ],
    "regex": "^/os-fping",
    "uri": "*/os-fping"
}
]
}
}

```

Example 3.102. Get limits: XML response

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

```
<limits xmlns:os-used-limits="http://docs.openstack.org/compute/ext/used_limits/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/common/api/v1.0">
  <rates>
    <rate regex=".*" uri="*">
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="POST" remaining="120" value="120"/>
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="PUT" remaining="120" value="120"/>
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="DELETE" remaining="120" value="120"/>
    </rate>
    <rate regex="^/servers" uri="*/servers">
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="POST" remaining="120" value="120"/>
    </rate>
    <rate regex=".*changes-since.*" uri="*changes-since*">
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="GET" remaining="120" value="120"/>
    </rate>
    <rate regex="^/os-fping" uri="*/os-fping">
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="GET" remaining="12" value="12"/>
    </rate>
  </rates>
  <absolute>
    <limit name="maxServerMeta" value="128"/>
    <limit name="maxPersonality" value="5"/>
    <limit name="maxImageMeta" value="128"/>
    <limit name="maxPersonalitySize" value="10240"/>
    <limit name="maxSecurityGroupRules" value="20"/>
    <limit name="maxTotalKeypairs" value="100"/>
    <limit name="totalRAMUsed" value="0"/>
    <limit name="totalInstancesUsed" value="0"/>
    <limit name="maxSecurityGroups" value="10"/>
    <limit name="totalFloatingIpsUsed" value="0"/>
    <limit name="maxTotalCores" value="20"/>
    <limit name="totalSecurityGroupsUsed" value="0"/>
    <limit name="maxTotalFloatingIps" value="10"/>
    <limit name="maxTotalInstances" value="10"/>
    <limit name="totalCoresUsed" value="0"/>
    <limit name="maxTotalRAMSize" value="51200"/>
  </absolute>
</limits>
```

This operation does not return a response body.

3.14. Limits with project usage for administrators (limits)

Extend limits to enable administrators to show the project usage for a specified customer project ID. Show information such as RAM or instance quotas usage.

Method	URI	Description
GET	/v2/{tenant_id}/limits/{tenant_id}	Enables administrators to get absolute and rate limit information, including information about currently used absolute limits, for a specified customer tenant ID.

3.14.1. Get customer limits

Method	URI	Description
GET	/v2/{tenant_id}/limits/{tenant_id}	Enables administrators to get absolute and rate limit information, including information about currently used absolute limits, for a specified customer tenant ID.

Normal response codes: 200

3.14.1.1. Request

This table shows the URI parameters for the get customer limits request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{tenant_id}	Uuid	The ID for the tenant for which an administrator wants to get quota information. Specify the customer tenant ID in the URI as tenant_id={tenant_id}.

This operation does not require a request body.

3.14.1.2. Response

Example 3.103. Used limits for admins: JSON response

```
{
  "limits": {
    "absolute": {
      "maxImageMeta": 128,
      "maxPersonality": 5,
      "maxPersonalitySize": 10240,
      "maxSecurityGroupRules": 20,
      "maxSecurityGroups": 10,
      "maxServerMeta": 128,
      "maxTotalCores": 20,
      "maxTotalFloatingIps": 10,
      "maxTotalInstances": 10,
      "maxTotalKeypairs": 100,
      "maxTotalRAMSize": 51200,
      "totalCoresUsed": 0,
      "totalInstancesUsed": 0,
      "totalRAMUsed": 0,
      "totalSecurityGroupsUsed": 0,
      "totalFloatingIpsUsed": 0
    },
    "rate": [
      {
        "limit": [
          {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "POST"
          }
        ],
        "next-available": "2012-11-27T17:24:52Z",
      }
    ]
  }
}
```

```

        "remaining": 120,
        "unit": "MINUTE",
        "value": 120,
        "verb": "PUT"
    },
    {
        "next-available": "2012-11-27T17:24:52Z",
        "remaining": 120,
        "unit": "MINUTE",
        "value": 120,
        "verb": "DELETE"
    }
],
"regex": ".*",
"uri": "*"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "POST"
        }
    ],
    "regex": "^/servers",
    "uri": "*/servers"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 120,
            "unit": "MINUTE",
            "value": 120,
            "verb": "GET"
        }
    ],
    "regex": ".*changes-since.*",
    "uri": "*changes-since*"
},
{
    "limit": [
        {
            "next-available": "2012-11-27T17:24:52Z",
            "remaining": 12,
            "unit": "MINUTE",
            "value": 12,
            "verb": "GET"
        }
    ],
    "regex": "^/os-fping",
    "uri": "*/os-fping"
}
]
}
}

```

Example 3.104. Used limits for admins: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<limits xmlns:os-used-limits="http://docs.openstack.org/compute/ext/
used_limits/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://
docs.openstack.org/common/api/v1.0">
  <rates>
    <rate regex=".*" uri="*">
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="POST"
remaining="120" value="120"/>
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="PUT"
remaining="120" value="120"/>
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="DELETE"
remaining="120" value="120"/>
    </rate>
    <rate regex="^/servers" uri="*/servers">
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="POST"
remaining="120" value="120"/>
    </rate>
    <rate regex=".*changes-since.*" uri="*changes-since*">
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="GET"
remaining="120" value="120"/>
    </rate>
    <rate regex="^/os-fping" uri="*/os-fping">
      <limit next-available="2012-11-27T17:24:53Z" unit="MINUTE" verb="GET"
remaining="12" value="12"/>
    </rate>
  </rates>
  <absolute>
    <limit name="maxServerMeta" value="128"/>
    <limit name="maxPersonality" value="5"/>
    <limit name="maxImageMeta" value="128"/>
    <limit name="maxPersonalitySize" value="10240"/>
    <limit name="maxSecurityGroupRules" value="20"/>
    <limit name="maxTotalKeypairs" value="100"/>
    <limit name="totalRAMUsed" value="0"/>
    <limit name="totalInstancesUsed" value="0"/>
    <limit name="maxSecurityGroups" value="10"/>
    <limit name="totalFloatingIpsUsed" value="0"/>
    <limit name="maxTotalCores" value="20"/>
    <limit name="totalSecurityGroupsUsed" value="0"/>
    <limit name="maxTotalFloatingIps" value="10"/>
    <limit name="maxTotalInstances" value="10"/>
    <limit name="totalCoresUsed" value="0"/>
    <limit name="maxTotalRAMSize" value="51200"/>
  </absolute>
</limits>
```

This operation does not return a response body.

3.15. Guest agents (os-agents)

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

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

3.15.1. List agents

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

Normal response codes: 200

3.15.1.1. Request

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

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

This operation does not require a request body.

3.15.1.2. Response

Example 3.105. List agents: JSON response

```
{
  "agents": [
    {
      "agent_id": "1",
      "architecture": "x86",
      "hypervisor": "hypervisor",
      "md5hash": "add6bb58e139be103324d04d82d8f545",
      "os": "os",
      "url": "xxxxxxxxxxxx",
      "version": "8.0"
    }
  ]
}
```

Example 3.106. List agents: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<agents>
  <agent url="xxxxxxxxxxxx" hypervisor="hypervisor" md5hash=
"add6bb58e139be103324d04d82d8f545" version="8.0" architecture="x86" os="os"
agent_id="1"/>
</agents>
```

This operation does not return a response body.

3.15.2. Create agent

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

Normal response codes: 200

3.15.2.1. Request

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

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

Example 3.107. Create agent: JSON request

```
{
  "agent": {
    "hypervisor": "hypervisor",
    "os": "os",
    "architecture": "x86",
    "version": "8.0",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "url": "xxxxxxxxxxxxx"
  }
}
```

Example 3.108. Create agent: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<agent>
  <hypervisor>hypervisor</hypervisor>
  <os>os</os>
  <architecture>x86</architecture>
  <version>8.0</version>
  <md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
  <url>xxxxxxxxxxxxx</url>
</agent>
```

This operation does not require a request body.

3.15.2.2. Response

Example 3.109. Create agent: JSON response

```
{
  "agent": {
    "agent_id": "1",
    "architecture": "x86",
    "hypervisor": "hypervisor",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "os": "os",
    "url": "xxxxxxxxxxxxx",
    "version": "8.0"
  }
}
```

```
}
```

Example 3.110. Create agent: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<agent>
  <url>xxxxxxxxxxxx</url>
  <hypervisor>hypervisor</hypervisor>
  <md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
  <version>8.0</version>
  <architecture>x86</architecture>
  <os>os</os>
  <agent_id>1</agent_id>
</agent>
```

This operation does not return a response body.

3.15.3. Delete agent

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

Normal response codes: 202

3.15.3.1. Request

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

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

This operation does not require a request body.

3.15.4. Update agent

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

Normal response codes: 200

3.15.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique ID of the tenant or account.
{id}	UUID	The unique ID associated with the agent.

Example 3.111. Update agent: JSON request

```
{
  "para": {
    "url": "xxx://xxxx/xxx/xxx",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "version": "7.0"
  }
}
```

Example 3.112. Update agent: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<para>
  <version>7.0</version>
  <url>xxx://xxxx/xxx/xxx</url>
  <md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
</para>
```

This operation does not require a request body.

3.15.4.2. Response

Example 3.113. Update agent: JSON response

```
{
  "agent": {
    "agent_id": "1",
    "md5hash": "add6bb58e139be103324d04d82d8f545",
    "url": "xxx://xxxx/xxx/xxx",
    "version": "7.0"
  }
}
```

Example 3.114. Update agent: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<agent>
  <url>xxx://xxxx/xxx/xxx</url>
```

```
<version>7.0</version>
<agent_id>1</agent_id>
<md5hash>add6bb58e139be103324d04d82d8f545</md5hash>
</agent>
```

This operation does not return a response body.

3.16. Host aggregates (os-aggregates)

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

Method	URI	Description
GET	/v2/{tenant_id}/os-aggregates	Lists all aggregates.
POST	/v2/{tenant_id}/os-aggregates	Creates an aggregate.
DELETE	/v2/{tenant_id}/os-aggregates/ {aggregate_id}	Deletes an aggregate.
GET	/v2/{tenant_id}/os-aggregates/ {aggregate_id}	Gets details about a specified aggregate.
PUT	/v2/{tenant_id}/os-aggregates/ {aggregate_id}	Updates the name, and optionally the availability zone, for a specified aggregate.
POST	/v2/{tenant_id}/os-aggregates/ {aggregate_id}/action	Sets metadata for an aggregate.
POST	/v2/{tenant_id}/os-aggregates/ {aggregate_id}/action	Adds a host to an aggregate.
POST	/v2/{tenant_id}/os-aggregates/ {aggregate_id}/action	Removes a host from an aggregate.

3.16.1. List aggregates

Method	URI	Description
GET	/v2/{tenant_id}/os-aggregates	Lists all aggregates.

Normal response codes: 200

3.16.1.1. Request

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

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

This operation does not require a request body.

3.16.1.2. Response

Example 3.115. List aggregates: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregates>
  <aggregate>
    <name>name</name>
    <availability_zone>nova</availability_zone>
    <deleted>False</deleted>
    <created_at>2012-11-16 06:22:25.935099</created_at>
    <updated_at>None</updated_at>
    <hosts/>
    <deleted_at>None</deleted_at>
    <id>1</id>
    <metadata>
      <availability_zone>nova</availability_zone>
    </metadata>
  </aggregate>
</aggregates>
```

Example 3.116. List aggregates: JSON response

```
{
  "aggregates": [
    {
      "availability_zone": "nova",
      "created_at": "2012-11-16T06:22:23.361359",
      "deleted": false,
      "deleted_at": null,
      "hosts": [],
      "id": 1,
      "metadata": {
        "availability_zone": "nova"
      },
      "name": "name",
      "updated_at": null
    }
  ]
}
```



```
}
```

This operation does not return a response body.

3.16.2. Create aggregate

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

Normal response codes: 200200

3.16.2.1. Request

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

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

Example 3.117. Create aggregate: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<aggregate name="name" availability_zone="nova" />
```

Example 3.118. Create aggregate: JSON request

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

This operation does not require a request body.

3.16.2.2. Response

Example 3.119. Create aggregate: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
  <name>name</name>
  <availability_zone>nova</availability_zone>
  <deleted>False</deleted>
  <created_at>2012-10-01 18:50:35.506667</created_at>
  <updated_at>None</updated_at>
  <deleted_at>None</deleted_at>
  <id>1</id>
</aggregate>
```

Example 3.120. Create aggregate: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2012-10-01T18:50:27.781065",
    "deleted": false,
    "deleted_at": null,
  }
}
```

```
    "id": 1,  
    "name": "name",  
    "updated_at": null  
  }  
}
```

This operation does not return a response body.

3.16.3. Delete aggregate

Method	URI	Description
DELETE	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Deletes an aggregate.

Normal response codes: 200

3.16.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

This operation does not require a request body.

3.16.4. Get aggregate details

Method	URI	Description
GET	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Gets details about a specified aggregate.

Normal response codes: 200200

3.16.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

This operation does not require a request body.

3.16.4.2. Response

Example 3.121. Get aggregate details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
  <name>name</name>
  <availability_zone>nova</availability_zone>
  <deleted>False</deleted>
  <created_at>2012-11-16 06:22:25.587739</created_at>
  <updated_at>None</updated_at>
  <hosts/>
  <deleted_at>None</deleted_at>
  <id>1</id>
  <metadata>
    <availability_zone>nova</availability_zone>
  </metadata>
</aggregate>
```

Example 3.122. Get aggregate details: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2012-11-16T06:22:23.032493",
    "deleted": false,
    "deleted_at": null,
    "hosts": [],
    "id": 1,
    "metadata": {
      "availability_zone": "nova"
    },
    "name": "name",
    "updated_at": null
  }
}
```

This operation does not return a response body.

3.16.5. Update aggregate

Method	URI	Description
PUT	/v2/{tenant_id}/os-aggregates/{aggregate_id}	Updates the name, and optionally the availability zone, for a specified aggregate.

Normal response codes: 200

3.16.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

Example 3.123. Update aggregate: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<aggregate name="newname" availability_zone="nova2" />
```

Example 3.124. Update aggregate: JSON request

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

This operation does not require a request body.

3.16.5.2. Response

Example 3.125. Update aggregate: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
  <name>newname</name>
  <availability_zone>nova2</availability_zone>
  <deleted>False</deleted>
  <created_at>2012-12-04 12:04:30.245284</created_at>
  <updated_at>2012-12-04 12:04:30.357795</updated_at>
  <hosts/>
  <deleted_at>None</deleted_at>
  <id>1</id>
  <metadata>
    <availability_zone>nova2</availability_zone>
  </metadata>
</aggregate>
```

Example 3.126. Update aggregate: JSON response

```
{
```

```
"aggregate": {
  "availability_zone": "nova2",
  "created_at": "2012-12-04T12:04:27.075065",
  "deleted": false,
  "deleted_at": null,
  "hosts": [],
  "id": 1,
  "metadata": {
    "availability_zone": "nova2"
  },
  "name": "newname",
  "updated_at": "2012-12-04T12:04:27.242597"
}
```

This operation does not return a response body.

3.16.6. Set aggregate metadata

Method	URI	Description
POST	/v2/{tenant_id}/os-aggregates/{aggregate_id}/action	Sets metadata for an aggregate.

Normal response codes: 200

3.16.6.1. Request

This table shows the URI parameters for the set aggregate metadata request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

Example 3.127. Set aggregate metadata: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<set_metadata>
  <metadata>
    <key>value</key>
  </metadata>
</set_metadata>
```

Example 3.128. Set aggregate metadata: JSON request

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

This operation does not require a request body.

3.16.6.2. Response

Example 3.129. Set aggregate metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
  <name>name</name>
  <availability_zone>nova</availability_zone>
  <deleted>False</deleted>
  <created_at>2012-11-16 06:22:24.864471</created_at>
  <updated_at>None</updated_at>
  <hosts/>
  <deleted_at>None</deleted_at>
  <id>1</id>
  <metadata>
```

```
<key>value</key>
<availability_zone>nova</availability_zone>
</metadata>
</aggregate>
```

Example 3.130. Set aggregate metadata: JSON response

```
{
  "aggregate": {
    "availability_zone": "nova",
    "created_at": "2012-11-16T06:22:22.342791",
    "deleted": false,
    "deleted_at": null,
    "hosts": [],
    "id": 1,
    "metadata": {
      "availability_zone": "nova",
      "key": "value"
    },
    "name": "name",
    "updated_at": null
  }
}
```

This operation does not return a response body.

3.16.7. Add host to aggregate

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

Normal response codes: 200

3.16.7.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

Example 3.131. Add host to aggregate: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<add_host host="7c9e00dbca5e4fb88538b021c0f933a5" />
```

Example 3.132. Add host to aggregate: JSON request

```
{
  "add_host": {
    "host": "581d29b9e3504d8a895caddb13839b15"
  }
}
```

This operation does not require a request body.

3.16.7.2. Response

Example 3.133. Add host to aggregate: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
  <name>name</name>
  <availability_zone>nova</availability_zone>
  <deleted>False</deleted>
  <created_at>2012-12-04 12:04:27.574038</created_at>
  <updated_at>None</updated_at>
  <hosts>
    <host>392adba19dd449179804eaff16ff4a97</host>
  </hosts>
  <deleted_at>None</deleted_at>
  <id>1</id>
  <metadata>
    <availability_zone>nova</availability_zone>
  </metadata>
</aggregate>
```

Example 3.134. Add host to aggregate: JSON response

```
{
```

```
"aggregate": {
  "availability_zone": "nova",
  "created_at": "2012-12-04T12:04:24.399784",
  "deleted": false,
  "deleted_at": null,
  "hosts": [
    "0438c6a4e8d841ad823b801d681f4680"
  ],
  "id": 1,
  "metadata": {
    "availability_zone": "nova"
  },
  "name": "name",
  "updated_at": null
}
```

This operation does not return a response body.

3.16.8. Remove host from aggregate

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

Normal response codes: 200

3.16.8.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{aggregate_id}	UUID	The ID associated with an aggregate.

Example 3.135. Remove host from aggregate: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<remove_host host="7c9e00dbca5e4fb88538b021c0f933a5" />
```

Example 3.136. Remove host from aggregate: JSON request

```
{
  "remove_host":
  {
    "host": "581d29b9e3504d8a895caddb13839b15"
  }
}
```

This operation does not require a request body.

3.16.8.2. Response

Example 3.137. Remove host from aggregate: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<aggregate>
  <name>name</name>
  <availability_zone>nova</availability_zone>
  <deleted>False</deleted>
  <created_at>2012-12-04 12:04:29.722109</created_at>
  <updated_at>None</updated_at>
  <hosts/>
  <deleted_at>None</deleted_at>
  <id>1</id>
  <metadata>
    <availability_zone>nova</availability_zone>
  </metadata>
</aggregate>
```

Example 3.138. Remove host from aggregate: JSON response

```
{
```

```

"aggregate": {
  "availability_zone": "nova",
  "created_at": "2012-12-04T12:04:26.557909",
  "deleted": false,
  "deleted_at": null,
  "hosts": [],
  "id": 1,
  "metadata": {
    "availability_zone": "nova"
  },
  "name": "name",
  "updated_at": null
}
}

```

This operation does not return a response body.

3.17. Attach interfaces (os-attach-interfaces)

Create, list, and get details for port interfaces.

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/os-attach-interfaces	Creates and uses a port interface to attach the port to a server instance.
GET	/v2/{tenant_id}/servers/{server_id}/os-attach-interfaces	Lists port interfaces.
GET	/v2/{tenant_id}/servers/{server_id}/os-attach-interfaces/{attachment_id}	Shows information about a specified port interface.

3.17.1. Create interface

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

Normal response codes: 202

3.17.1.1. Request

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

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

Example 3.139. Create interface: JSON request

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

Example 3.140. Create interface: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<interfaceAttachment>
  <port_id>ce531f90-199f-48c0-816c-13e38010b442</port_id>
</interfaceAttachment>
```

This operation does not require a request body.

3.17.1.2. Response

Example 3.141. Create interface: JSON response

```
{
  "interfaceAttachment": {
    "fixed_ips": [
      {
        "ip_address": "192.168.1.1",
        "subnet_id": "f8a6e8f8-c2ec-497c-9f23-da9616de54ef"
      }
    ],
    "mac_addr": "fa:16:3e:4c:2c:30",
    "net_id": "3cb9bc59-5699-4588-a4b1-b87f96708bc6",
    "port_id": "ce531f90-199f-48c0-816c-13e38010b442",
    "port_state": "ACTIVE"
  }
}
```

Example 3.142. Create interface: XML response

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

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

This operation does not return a response body.

3.17.2. List interfaces

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-attach-interfaces	Lists port interfaces.

Normal response codes: 202

3.17.2.1. Request

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

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

This operation does not require a request body.

3.17.2.2. Response

Example 3.143. List interfaces: JSON response

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

Example 3.144. List interfaces: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<interfaceAttachments>
  <interfaceAttachment>
    <port_state>ACTIVE</port_state>
    <fixed_ips>
      <fixed_ip>
        <subnet_id>f8a6e8f8-c2ec-497c-9f23-da9616de54ef</subnet_id>
        <ip_address>192.168.1.3</ip_address>
      </fixed_ip>
    </fixed_ips>
    <port_id>ce531f90-199f-48c0-816c-13e38010b442</port_id>
    <net_id>3cb9bc59-5699-4588-a4b1-b87f96708bc6</net_id>
    <mac_addr>fa:16:3e:4c:2c:30</mac_addr>
  </interfaceAttachment>
</interfaceAttachments>
```

```
</interfaceAttachments>
```

This operation does not return a response body.

3.17.3. Show attached interface information

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-attach-interfaces/{attachment_id}	Shows information about a specified port interface.

Normal response codes: 202

3.17.3.1. Request

This table shows the URI parameters for the show attached interface information request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.
{attachment_id}	UUID	The interface ID.

This operation does not require a request body.

3.17.3.2. Response

Example 3.145. Show attached interface information: JSON response

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

Example 3.146. Show attached interface information: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<interfaceAttachment>
  <port_state>ACTIVE</port_state>
  <fixed_ips>
    <fixed_ip>
      <subnet_id>b6e47749-6bf0-4d6e-ae4b-ba6b5e238510</subnet_id>
      <ip_address>192.168.123.131</ip_address>
    </fixed_ip>
  </fixed_ips>
  <port_id>89e64f2e-86bd-4c19-9155-4548b36fdbcb2</port_id>
  <net_id>a9efd207-2c1a-4cdd-a296-d3c7c3211302</net_id>
  <mac_addr>fa:16:3e:a4:1c:12</mac_addr>
</interfaceAttachment>
```

This operation does not return a response body.

3.18. Root certificates (os-certificates)

Creates and show details for a root certificate.

Method	URI	Description
POST	/v2/{tenant_id}/os-certificates	Creates a root certificate.
GET	/v2/{tenant_id}/os-certificates	Shows details for a root certificate owned by a specified tenant ID.

3.18.1. Create root certificate

Method	URI	Description
POST	/v2/{tenant_id}/os-certificates	Creates a root certificate.

Normal response codes: 202

3.18.1.1. Request

This table shows the URI parameters for the create root certificate request:

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

This operation does not require a request body.

3.18.1.2. Response

Example 3.147. Create root certificate: JSON response

```
{
  "certificate": {
    "data": "Certificate:\n      Data:\n      Version: 1 (0x0)\n
Serial Number: 23 (0x17)\n      Signature Algorithm: md5WithRSAEncryption
\n      Issuer: O=NOVA ROOT, L=Mountain View, ST=California, C=US\n
Validity\n      Not Before: Oct  2 19:31:45 2012 GMT\n
Not After : Oct  2 19:31:45 2013 GMT\n      Subject: C=US, ST=California,
O=OpenStack, OU=NovaDev, CN=openstack-fake-2012-10-02T19:31:45Z\n
Subject Public Key Info:\n      Public Key Algorithm: rsaEncryption
\n      RSA Public Key: (1024 bit)\n      Modulus (1024
bit):\n      00:b8:87:67:7a:de:28:ed:f6:5d:1f:20:14:58:df:
\n      b0:f7:62:3d:85:61:a8:c2:31:49:5f:b5:2a:07:34:
\n      0e:25:13:0d:2e:4d:79:c7:26:36:dd:d2:3e:c7:36:
\n      54:80:0d:67:80:32:e6:a8:48:33:69:ec:22:2c:5c:
\n      cb:7a:88:0f:c0:48:de:67:14:54:d9:94:b4:6a:23:
\n      36:28:23:44:47:8a:24:89:8e:f4:86:77:59:f8:62:
\n      b6:ab:d5:e0:bc:b4:ad:d1:95:dd:59:a3:aa:e3:ea:
\n      d3:ae:23:17:c5:54:96:a3:25:56:72:90:20:07:8c:
\n      63:4d:be:e9:60:7e:10:57:17\n
Exponent: 65537 (0x10001)\n      Signature Algorithm: md5WithRSAEncryption
\n      32:82:ff:8b:92:0e:8d:9c:6b:ce:7e:fe:34:16:2a:4c:47:4f:
\n      c7:28:a2:33:1e:48:56:2e:4b:e8:e8:e3:48:b1:3d:a3:43:21:
\n      ef:83:e7:df:e2:10:91:7e:9a:c0:4d:1e:96:68:2b:b9:f7:84:
\n      7f:ec:84:8a:bf:bc:5e:50:05:d9:ce:4a:1a:bf:d2:bf:0c:d1:
\n      7e:ec:64:c3:a5:37:78:a3:a6:2b:a1:b7:1c:cc:c8:b9:78:61:
\n      98:50:3c:e6:28:34:f1:0e:62:bb:b5:d7:a1:dd:1f:38:c6:0d:
\n      58:9f:81:67:ff:9c:32:fc:52:7e:6d:8c:91:43:49:fe:e3:48:
\n      bb:40\n-----BEGIN CERTIFICATE-----\n
nMIICMzCCAzwCARcwDQYJKoZIhvcNAQEEBQAwtjESMBAGA1UEChMJTk9WQSBST09U\
nMRYwFAYDVQQHEw1Nb3VudGFpbmIBwWV3MRMwEQYDVQQIEwpDYWxpZm9ybmlhMQsw\
nCQYDVQQGEwJVeZAEFw0xMjEwMDIwMDIwMDIwMDIwMDIwMDIwMDIwMDIwMDIwMDIw\
nBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRlweEYDVQQKEw1PcGVuU3Rl\
nY2sxEDA0BgNVBAsTB05vdmFEZXYxLDAqBgNVBAMTI29wZW5zdGFjay1mYWt1LlRl\
nMTItMTAtMDJUMTk2MzE6NDVAMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC4\
nh2d63ijt910fIBRY37D3Yj2FYajcMULftSoHNA4lEw0uTXnHJjbd0j7HNlSADWeA\
```

```
nMuoSDNp7CIsXMT6iA/ASN5nFFTZlLRqIzYoI0RHiiSJjvSGdln4YrarleC8tK3R\
nld1Zo6rj6tOuIxfFVJajJVZykCAHjGNNvulgfHbXfWIDAQABMA0GCSqGSIB3DQEB\
nBAUAA4GBADKC/4uSDo2ca85+/jQWkKxHT8cooJmeSFYuS+jo40ixPaNDIE+d59/i\
nEJF+msBNHpZoK7n3hH/shIq/vF5QBdnOShq/0r8M0X7sZM0lN3ijpiuhtxzMyLl4\
nYZhQPOYoNPEOYru116HdHzjGDVifgWf/nDL8Un5tjJFDSf7jSlTA\n-----END
CERTIFICATE-----\n",
    "private_key": "-----BEGIN RSA PRIVATE KEY-----\
nMIICXAIBAAKBgQC4h2d63ijt9l0fIBRY37D3Yj2FYyajCMUlfTsoHNA4lEw0uTXnH\
nJjbd0j7HNlSADWeAMuoSDNp7CIsXMT6iA/ASN5nFFTZlLRqIzYoI0RHiiSJjvSG\
ndln4YrarleC8tK3Rld1Zo6rj6tOuIxfFVJajJVZykCAHjGNNvulgfHbXfWIDAQAB\
nAoGBAijfxx4YU/vO1lwUC40wys92q30YcPk6XdakJryZHDTb4NcLmNzjt6bqIK7b\
n2enyB2fMwDNRWvGiueZ2HmiRLDyOGsAVdEsHvL4qbr9EZGTqC8Qxx+zTevWWf6pB\
nFlzxxzBXNQDFZdf9kVsSLCkbMHITnWlk4MrM++9gfcO3WrfEhAkEA4nd8TyCCZazq\
nKMOQwFLTNaiVLeTXCtvGopl4ZNIkYZlqI3KDXb2wbAyArFuERlotxFlylXpwtlMo\
nSlI/C/sYqwJBANCX1sdfRjQ8DpdP44ThWqOkWFLB9rBiwyBt8746fX8amwr8eyz\
nH44/z5GT/Vyp8qFsJkuDzeP93eeDnr2qe0UCPlzipRnPO6x4P5J4o+Y+EmLvwkAQ\
nnCLYAaCvUbILHrbq2Z2wWjEYnEO03RHud2xjkGH4TgcBMTmW4e+ZzEIduwJACnIw\
nLVfWBbG5QVac3EC021EVoz9XbUnk4Eu2usS4Yrs7USN6QBJQWD1VlcKfG6h3ICJh\
nleKJ4wsJm9h5kKH9yQJBAN8CaX223MlTSuBOVuIOwNA+09iLfx4UCLiH1fGMKDpe\
nxVcmkM3qCnTqNxrAPSFdT9IyB3IXiaLWbvzl7MfiOwQ=\n-----END RSA PRIVATE KEY-----\
n"
    }
}
```

Example 3.148. Create root certificate: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<certificate
  private_key="-----BEGIN RSA PRIVATE KEY-----&
#10;MIICXAIBAAKBgQC4h2d63ijt9l0fIBRY37D3Yj2FYyajCMUlfTsoHNA4lEw0uTXnH&
#10;Jjbd0j7HNlSADWeAMuoSDNp7CIsXMT6iA/ASN5nFFTZlLRqIzYoI0RHiiSJjvSG&
#10;dln4YrarleC8tK3Rld1Zo6rj6tOuIxfFVJajJVZykCAHjGNNvulgfHbXfWIDAQAB&
#10;AoGBAijfxx4YU/vO1lwUC40wys92q30YcPk6XdakJryZHDTb4NcLmNzjt6bqIK7b&
#10;2enyB2fMwDNRWvGiueZ2HmiRLDyOGsAVdEsHvL4qbr9EZGTqC8Qxx+zTevWWf6pB&
#10;FlzxxzBXNQDFZdf9kVsSLCkbMHITnWlk4MrM++9gfcO3WrfEhAkEA4nd8TyCCZazq&
#10;KMOQwFLTNaiVLeTXCtvGopl4ZNIkYZlqI3KDXb2wbAyArFuERlotxFlylXpwtlMo&
#10;SlI/C/sYqwJBANCX1sdfRjQ8DpdP44ThWqOkWFLB9rBiwyBt8746fX8amwr8eyz&
#10;H44/z5GT/Vyp8qFsJkuDzeP93eeDnr2qe0UCPlzipRnPO6x4P5J4o+Y+EmLvwkAQ&
#10;nCLYAaCvUbILHrbq2Z2wWjEYnEO03RHud2xjkGH4TgcBMTmW4e+ZzEIduwJACnIw&
#10;LVfWBbG5QVac3EC021EVoz9XbUnk4Eu2usS4Yrs7USN6QBJQWD1VlcKfG6h3ICJh&
#10;leKJ4wsJm9h5kKH9yQJBAN8CaX223MlTSuBOVuIOwNA+09iLfx4UCLiH1fGMKDpe&
#10;xVcmkM3qCnTqNxrAPSFdT9IyB3IXiaLWbvzl7MfiOwQ=&#10;-----END RSA PRIVATE
KEY-----&#10;"
  data="Certificate:&#10;      Data:&#10;          Version: 1 (0x0)&
#10;      Serial Number: 23 (0x17)&#10;          Signature Algorithm:
md5WithRSAEncryption&#10;      Issuer: O=NOVA ROOT, L=Mountain View,
ST=California, C=US&#10;          Validity&#10;          Not Before: Oct
2 19:31:45 2012 GMT&#10;          Not After : Oct 2 19:31:45 2013
GMT&#10;          Subject: C=US, ST=California, O=OpenStack, OU=NovaDev,
CN=openstack-fake-2012-10-02T19:31:45Z&#10;          Subject Public Key
Info:&#10;          Public Key Algorithm: rsaEncryption&#10;
RSA Public Key: (1024 bit)&#10;          Modulus (1024 bit):&
#10;          00:b8:87:67:7a:de:28:ed:f6:5d:1f:20:14:58:df:&
#10;          b0:f7:62:3d:85:61:a8:c2:31:49:5f:b5:2a:07:34:&
#10;          0e:25:13:0d:2e:4d:79:c7:26:36:dd:d2:3e:c7:36:&
#10;          54:80:0d:67:80:32:e6:a8:48:33:69:ec:22:c2:5c:&
#10;          cb:7a:88:0f:c0:48:de:67:14:54:d9:94:b4:6a:23:&
#10;          36:28:23:44:47:8a:24:89:8e:f4:86:77:59:f8:62:&
#10;          b6:ab:d5:e0:bc:b4:ad:d1:95:dd:59:a3:aa:e3:ea:&
#10;          d3:ae:23:17:c5:54:96:a3:25:56:72:90:20:07:8c:&
```

```
#10;          63:4d:be:e9:60:7e:10:57:17&#10;
  Exponent: 65537 (0x10001)&#10;    Signature Algorithm: md5WithRSAEncryption&
#10;          32:82:ff:8b:92:0e:8d:9c:6b:ce:7e:fe:34:16:2a:4c:47:4f:&
#10;          c7:28:a2:33:1e:48:56:2e:4b:e8:e8:e3:48:b1:3d:a3:43:21:&
#10;          ef:83:e7:df:e2:10:91:7e:9a:c0:4d:1e:96:68:2b:b9:f7:84:&
#10;          7f:ec:84:8a:bf:bc:5e:50:05:d9:ce:4a:1a:bf:d2:bf:0c:d1:&
#10;          7e:ec:64:c3:a5:37:78:a3:a6:2b:a1:b7:1c:cc:c8:b9:78:61:&
#10;          98:50:3c:e6:28:34:f1:0e:62:bb:b5:d7:a1:dd:1f:38:c6:0d:&
#10;          58:9f:81:67:ff:9c:32:fc:52:7e:6d:8c:91:43:49:fe:e3:48:&
#10;          bb:40&#10;-----BEGIN CERTIFICATE-----&
#10;MIICMzCCAzwCARcwDQYJKoZIhvcNAQEEBQAwTjESMBAGAlUEChMJTk9WQSBST09U&
#10;MRYwFAYDVQQHEw1Nb3VudGFpbWV3MRMwEQYDVQQLIEwpcDYWxpZm9ybmlhMQsw&
#10;CQYDVQQGEwJVUzAeFw0xMjEwMDIxOTMxNDVaFw0xMzEwMDIxOTMxNDVAMHYxCzAJ&
#10;BgNVBAYTAlVTMRMwEQYDVQQLIEwpcDYWxpZm9ybmlhMRIwEAYDVQQKEw1PcGVuU3Rh&
#10;Y2sxEDA0BgNVBAsTB05vdmFEZXYxLDAqBgNVBAMTI29wZW5zdGFjaylmYWtlLTlIw&
#10;MTItMTAtMDJUMTk6MzE6NDVAMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC4&
#10;h2d63ijt9l0fIBRY37D3Yj2FYajCMU1ftSoHNA4lEw0uTXnHJjbd0j7HN1SADWeA&
#10;MuaosDNp7CIsXmt6ia/ASN5nFFTZLLRqIzYoIORHisiSjvSGdl4YrarleC8tK3R&
#10;ldlZo6rj6tOuxfFVJajJVZykCAHjGNNvulgfhBXFwIDAQABMA0GCSqGSIb3DQEB&
#10;BAUAA4GBADKC/4uSDo2ca85+/jQWKkxHT8coojMeSFYuS+jo40ixPaNDIe+D59/i&
#10;EJF+msBNHpZok7n3hH/shIq/vF5QBdnOShq/Or8M0X7sZM0LN3ijpiuhtxzMyLl4&
#10;YZhQPOYonPEOYru116HdHzjGDVifgWf/nDL8Un5tjJFDSf7jSLtA&#10;-----END
CERTIFICATE-----&#10;"/>
```

This operation does not return a response body.

3.18.2. Show root certificate details

Method	URI	Description
GET	/v2/{tenant_id}/os-certificates	Shows details for a root certificate owned by a specified tenant ID.

Normal response codes: 200

3.18.2.1. Request

This table shows the URI parameters for the show root certificate details request:

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

This operation does not require a request body.

3.18.2.2. Response

Example 3.149. Show root certificate details: JSON response

```
{
  "certificate": {
    "data": "-----BEGIN CERTIFICATE-----\n
nMIICyzCCAjSgAwIBAgIJAIJ/UoFWKooOUMA0GCSqGSIb3DQEBAUAME4xEjAQBgNV\
nBAoTCU5PVkEgUk9PVDEWMBQGA1UEBxMNTW91bnRhaW4gVmllZETMBEgAlUECBMK\
nQ2FsaWZvcn5pYTELMAkGA1UEBhMCMVVMwHhcNMTIxMDAyMTg1NzQ1WhcNMTIxMDAy\
nMTg1NzQ1WjBOMRIwEAYDVQQKEw1OT1ZBIFJPT1QxZjAUBgNVBACjAUBgNVBAYTA1VTMIGfMA0GCSqG\
nSIB3DQEBAQUAA4GNADCBiQKBggQCCg0Bn8WSqbJF3QNTZUxo1TzmFBxuvhjZLKbnQ\
nIiShdVIWUK7RC8frq8FJI7dgJNmvkIBn9njABWDoZmurQRCzD65yCSbUc4R2ea5H\
nIK4wQIui0CJykvMBNjAe3bzztVVs8/ccDTs jtqq3F/KeQkKzQVfSWBrJSmYtG5tO\
nG+dOsWIDAQABo4GwMIGtMAwGA1UdEwQFMAMBAf8wHQYDVR0OBBYEFCljRfANosA/\
n9mHug0io7Lt83FtaMH4GAlUdIwR3MHWAFCljRfANosA/9mHug0io7Lt83FtaoVKk\
nUDBOMRIwEAYDVQQKEw1OT1ZBIFJPT1QxZjAUBgNVBACjAUBgNVBAYTA1VTggkAgn9SgVYqg5QwDQYJ\
nEzARBgNVBAGTCkNhbgG1mb3JuaWEwXzAUBgNVBAYTA1VTggkAgn9SgVYqg5QwDQYJ\
nKoZlHvcNAQEEBQADgYEAEBpJ00lpKCh5omwfAwAfFg1ml4h/FJiCH3PETmOCc+3l\
nCtWTBd4MG8AoH7A3PU2JKAGVQ5XWo6+ihpW1RgfQpCnIoI6vIeGcws+rSLnlzULt\
nIvfCjPrG7iQdR3jZGt3295behtPlGsCqipJEulOkOaEIs8iLlXgSOG94Mkw1b4Q=\n
-----END
CERTIFICATE-----\n",
    "private_key": null
  }
}
```

Example 3.150. Show root certificate details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<certificate private_key="None"
  data="-----BEGIN CERTIFICATE-----&#10;nMIICyzCCAjSgAwIBAgIJAIJ/
UoFWKooOUMA0GCSqGSIb3DQEBAUAME4xEjAQBgNV&
#10;nBAoTCU5PVkEgUk9PVDEWMBQGA1UEBxMNTW91bnRhaW4gVmllZETMBEgAlUECBMK&
#10;nQ2FsaWZvcn5pYTELMAkGA1UEBhMCMVVMwHhcNMTIxMDAyMTg1NzQ1WhcNMTIxMDAy&
#10;nMTg1NzQ1WjBOMRIwEAYDVQQKEw1OT1ZBIFJPT1QxZjAUBgNVBACjAUBgNVBAYTA1VTMIGfMA0GCSqG&
#10;nSIB3DQEBAQUAA4GNADCBiQKBggQCCg0Bn8WSqbJF3QNTZUxo1TzmFBxuvhjZLKbnQ&
```



```
#10;IiShdVIWUK7RC8frq8FJI7dgJNmvkIBn9njABWDoZmurQRCzD65yCSbUc4R2ea5H&
#10;IK4wQIui0CJykvMBNjAe3bzztVV8/ccDTsjtqq3F/KeQkKzQVfSWBrJSmYtG5t0&
#10;G+dOSwIDAQABo4GwMIGtMAwGA1UdEwQFMAMBAf8wHQYDVR0OBByEFC1jRfaNOsA/&
#10;9mHuq0io7Lt83FtaMH4GA1UdIwR3MHwAFCljRfaNOsA/9mHuq0io7Lt83FtaoVKk&
#10;UDBOMRIwEAYDVQKEwLOT1ZBIFJPT1QxFjAUBgNVBAcTDU1vdW50YWluIFZpZXCx&
#10;EzARBgNVBAgTCkNhbg1mb3JuaWEeXzAjbG9NVBAYTAlVTggkAgn9SgVYqg5QwDQYJ&
#10;KozIhvcNAQEEBQADgYEAEbpJO0lpKCh5omwfAwAfFglml4h/FJiCH3PETmOCc+3l&
#10;CtWTBd4MG8AoH7A3PU2JKAGVQ5XWo6+ihpW1RgfQpCnloI6vIeGcws+rSLnlzULt&
#10;IvfCJpRg7iQdR3jzGt3295behtPlGsCqipJEulOkOaEIs8iLlXgSOG94MkwlB4Q=&#10;-----
END CERTIFICATE-----&#10;"/>
```

This operation does not return a response body.

3.19. Cloudpipe (os-cloudpipe)

Manage virtual VPNs for projects.

Method	URI	Description
GET	/v2/{tenant_id}/os-cloudpipe	Lists cloudpipes.
POST	/v2/{tenant_id}/os-cloudpipe{?project_id}	Creates a cloudpipe.
POST	/v2/{tenant_id}/os-cloudpipe/configure-project	Updates the virtual private network (VPN) IP address and port for a specified cloudpipe instance.

3.19.1. List cloudpipes

Method	URI	Description
GET	/v2/{tenant_id}/os-cloudpipe	Lists cloudpipes.

Normal response codes: 200

3.19.1.1. Request

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

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

This operation does not require a request body.

3.19.1.2. Response

Example 3.151. 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": "cloudpipe-fa1765bd-a352-49c7-a6b7-8ee108a3cb0c",
      "public_ip": "127.0.0.1",
      "public_port": 22,
      "state": "down"
    }
  ]
}
```

Example 3.152. List cloudpipes: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<cloudpipes>
  <cloudpipe>
    <public_ip>127.0.0.1</public_ip>
    <created_at>2012-09-25T18:18:49Z</created_at>
    <public_port>22</public_port>
    <state>down</state>
    <instance_id>1a17f615-343b-430f-976a-457c029eddb7</instance_id>
    <internal_ip>192.168.0.3</internal_ip>
    <project_id>cloudpipe-6405f2ca-caf9-493b-a1f6-e55f595d75ab</project_id>
  </cloudpipe>
</cloudpipes>
```

This operation does not return a response body.

3.19.2. Create cloudpipe

Method	URI	Description
POST	/v2/{tenant_id}/os-cloudpipe{?project_id}	Creates a cloudpipe.

Normal response codes: 200

3.19.2.1. Request

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

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

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

Name	Type	Description
project_id	String (Optional)	Creates the cloudpipe for the specified project ID. If omitted, the project ID defaults to the calling tenant.

Example 3.153. Create cloudpipe: JSON request

```
{
  "cloudpipe": {
    "project_id": "cloudpipe-059f21e3-c20e-4efc-9e7a-eba2ab3c6f9a"
  }
}
```

Example 3.154. Create cloudpipe: XML request

```
<cloudpipe
  project_id="cloudpipe-6405f2ca-caf9-493b-a1f6-e55f595d75ab"
/>
```

This operation does not require a request body.

3.19.2.2. Response

Example 3.155. Create cloudpipe: JSON response

```
{
  "instance_id": "1e9b8425-34af-488e-b969-4d46f4a6382e"
}
```

Example 3.156. Create cloudpipe: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<cloudpipe>
  <instance_id>1a17f615-343b-430f-976a-457c029eddb7</instance_id>
</cloudpipe>
```

This operation does not return a response body.

3.19.3. Update cloudpipe

Method	URI	Description
POST	/v2/{tenant_id}/os-cloudpipe/ configure-project	Updates the virtual private network (VPN) IP address and port for a specified cloudpipe instance.

Normal response codes: 202

3.19.3.1. Request

This table shows the header parameters for the update cloudpipe request:

Name	Type	Description
vpn_ip	String <i>(Required)</i>	The VPN IP address.
vpn_port	String <i>(Required)</i>	The VPN port.

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

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

Example 3.157. Update cloudpipe: JSON request

```
{
  "configure_project": {
    "vpn_ip": "192.168.1.1",
    "vpn_port": "2000"
  }
}
```

Example 3.158. Update cloudpipe: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<configure_project>
  <vpn_ip>192.168.1.1</vpn_ip>
  <vpn_port>2000</vpn_port>
</configure_project>
```

This operation does not require a request body.

3.20. Coverage reports (os-coverage)

Method	URI	Description
POST	/v2/{tenant_id}/os-coverage/action	Generates a coverage report.
POST	/v2/{tenant_id}/os-coverage/action	Starts Nova coverage reporting.
POST	/v2/{tenant_id}/os-coverage/action	Starts coverage reporting for all services.
POST	/v2/{tenant_id}/os-coverage/action	Stops coverage reporting.

3.20.1. Get coverage report

Method	URI	Description
POST	/v2/{tenant_id}/os-coverage/action	Generates a coverage report.

Normal response codes: 200

3.20.1.1. Request

This table shows the URI parameters for the get coverage report request:

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

Example 3.159. Start combined report: JSON request

```
{
  "report": {
    "xml": true,
    "file": "report"
  }
}
```

Example 3.160. Start combined report: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<report>
  <file>report</file>
  <xml>True</xml>
</report>
```

This operation does not require a request body.

3.20.1.2. Response

Example 3.161. Get coverage report: JSON response

```
{
  "path": "/tmp/tmp6kdYaa/nova-coverage_TOTUbz/report"
}
```

Example 3.162. Get coverage report: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<path>/tmp/tmp4j87bp/nova-coverage_7ViTA7/report</path>
```

This operation does not return a response body.

3.20.2. Start coverage report

Method	URI	Description
POST	/v2/{tenant_id}/os-coverage/action	Starts Nova coverage reporting.

Normal response codes: 202

3.20.2.1. Request

This table shows the URI parameters for the start coverage report request:

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

Example 3.163. Start coverage report: JSON request

```
{
  "start" : {
  }
}
```

Example 3.164. Start coverage report: XML request

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

This operation does not require a request body.

3.20.3. Start combined coverage report

Method	URI	Description
POST	/v2/{tenant_id}/os-coverage/action	Starts coverage reporting for all services.

All reports are combined into a single report.

Normal response codes: 200

3.20.3.1. Request

This table shows the URI parameters for the start combined coverage report request:

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

Example 3.165. Start combined report: JSON request

```
{
  "start" : {
    "combine": true
  }
}
```

Example 3.166. Start combined report: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<start>
  <combine>True</combine>
</start>
```

This operation does not require a request body.

3.20.4. Stop coverage report

Method	URI	Description
POST	/v2/{tenant_id}/os-coverage/action	Stops coverage reporting.

Normal response codes: 202

3.20.4.1. Request

This table shows the URI parameters for the stop coverage report request:

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

Example 3.167. Stop coverage report: JSON request

```
{
  "stop" : {
  }
}
```

Example 3.168. Stop coverage report: XML request

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

This operation does not require a request body.

3.20.4.2. Response

Example 3.169. Stop report: JSON response

```
{
  "path": "/tmp/tmpua9HvB/nova-coverage_rs2CaS"
}
```

Example 3.170. Stop report: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<path>/tmp/tmpCLve38/nova-coverage_GJ4BZ_</path>
```

This operation does not return a response body.

3.21. Fixed IPs (os-fixed-ips)

Shows data for a specified fixed IP, such as host name, CIDR, and address. Also, reserve or free a fixed IP.

Method	URI	Description
GET	/v2/{tenant_id}/os-fixed-ips/{fixed_ip}	Shows information for a specified fixed IP address.

Method	URI	Description
POST	/v2/{tenant_id}/os-fixed-ips/ {fixed_ip}/action	Reserves or releases a fixed IP.

3.21.1. Show fixed IP information

Method	URI	Description
GET	/v2/{tenant_id}/os-fixed-ips/{fixed_ip}	Shows information for a specified fixed IP address.

Normal response codes: 200

3.21.1.1. Request

This table shows the URI parameters for the show fixed ip information request:

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

This operation does not require a request body.

3.21.1.2. Response

Example 3.171. Show fixed IP information: JSON response

```
{
  "fixed_ip": {
    "address": "192.168.1.1",
    "cidr": "192.168.1.0/24",
    "host": "host",
    "hostname": "openstack"
  }
}
```

Example 3.172. Show fixed IP information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<fixed_ip>
  <cidr>192.168.1.0/24</cidr>
  <hostname>openstack</hostname>
  <host>host</host>
  <address>192.168.1.1</address>
</fixed_ip>
```

This operation does not return a response body.

3.21.2. Reserve or release a fixed IP

Method	URI	Description
POST	/v2/{tenant_id}/os-fixed-ips/{fixed_ip}/action	Reserves or releases a fixed IP.

To reserve a fixed IP address, specify `reserve` in the request body. To release a fixed IP address, specify `unreserve` in the request body.

Normal response codes: 202

3.21.2.1. Request

This table shows the URI parameters for the reserve or release a fixed ip request:

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

Example 3.173. Reserve or release a fixed IP: JSON request

```
{
  "reserve": "None"
}
```

Example 3.174. Reserve or release a fixed IP: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<reserve>None</reserve>
```

This operation does not require a request body.

3.22. Floating IP DNS records (os-floating-ip-dns)

Manage DNS records associated with IP addresses allocated by the floating IPs extension. Requests are dispatched to a DNS driver selected at startup.

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ip-dns	Lists registered DNS domains published by the DNS drivers.
PUT	/v2/{tenant_id}/os-floating-ip-dns/{domain}	Creates or updates a DNS domain.
DELETE	/v2/{tenant_id}/os-floating-ip-dns/{domain}	Deletes a DNS domain and all associated host entries.
PUT	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Creates or updates a DNS entry.
GET	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Finds a unique DNS entry for a specified domain and name.
DELETE	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Deletes a specified DNS entry.
GET	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}	Lists DNS entries for a specified domain and IP.

3.22.1. List DNS domains

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ip-dns	Lists registered DNS domains published by the DNS drivers.

Normal response codes: 200

3.22.1.1. Request

This table shows the URI parameters for the list dns domains request:

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

This operation does not require a request body.

3.22.1.2. Response

Example 3.175. List DNS domains: JSON response

```
{
  "domain_entries": [
    {
      "availability_zone": null,
      "domain": "domain1.example.org",
      "project": "project1",
      "scope": "public"
    }
  ]
}
```

Example 3.176. List DNS domains: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<domain_entries>
  <domain_entry project="project1" scope="public" domain="domain1.example.org"
  availability_zone="None"/>
</domain_entries>
```

This operation does not return a response body.

3.22.2. Create or update DNS domain

Method	URI	Description
PUT	/v2/{tenant_id}/os-floating-ip-dns/{domain}	Creates or updates a DNS domain.

Normal response codes: 200

3.22.2.1. Request

This table shows the URI parameters for the create or update dns domain request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.

Example 3.177. Create or update DNS domain: JSON request

```
{
  "domain_entry": {
    "domain": "domain1.example.org",
    "scope": "public",
    "project": "project1"
  }
}
```

Example 3.178. Create or update DNS domain: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<domain_entry>
  <domain>domain1.example.org</domain>
  <scope>public</scope>
  <project>project1</project>
</domain_entry>
```

This operation does not require a request body.

3.22.2.2. Response

Example 3.179. Create or update DNS domain: JSON response

```
{
  "domain_entry": {
    "availability_zone": null,
    "domain": "domain1.example.org",
    "project": "project1",
    "scope": "public"
  }
}
```

Example 3.180. Create or update DNS domain: XML response

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

```
<domain_entry project="project1" scope="public" domain="domain1.example.org"  
availability_zone="None" />
```

This operation does not return a response body.

3.22.3. Delete DNS domain

Method	URI	Description
DELETE	/v2/{tenant_id}/os-floating-ip-dns/{domain}	Deletes a DNS domain and all associated host entries.

Normal response codes: 200

3.22.3.1. Request

This table shows the URI parameters for the delete dns domain request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.

This operation does not require a request body.

3.22.4. Create or update DNS entry

Method	URI	Description
PUT	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Creates or updates a DNS entry.

Normal response codes: 200

3.22.4.1. Request

This table shows the URI parameters for the create or update dns entry request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.
{name}	String	DNS entry name assigned under a domain.

Example 3.181. Create or update DNS entry: JSON request

```
{
  "dns_entry": {
    "ip": "192.168.53.11",
    "dns_type": "A"
  }
}
```

Example 3.182. Create or update DNS entry: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<dns_entry>
  <ip>192.168.53.11</ip>
  <dns_type>A</dns_type>
</dns_entry>
```

This operation does not require a request body.

3.22.4.2. Response

Example 3.183. Create or update DNS entry: JSON response

```
{
  "dns_entry": {
    "domain": "domain1.example.org",
    "id": null,
    "ip": "192.168.1.1",
    "name": "instancel",
    "type": "A"
  }
}
```

Example 3.184. Create or update DNS entry: XML response

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



```
<dns_entry ip="192.168.1.1" domain="domain1.example.org" type="A" id="None"  
name="instance1"/>
```

This operation does not return a response body.

3.22.5. Find unique DNS entry

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Finds a unique DNS entry for a specified domain and name.

Normal response codes: 200

3.22.5.1. Request

This table shows the URI parameters for the find unique dns entry request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.
{name}	String	DNS entry name assigned under a domain.

This operation does not require a request body.

3.22.5.2. Response

Example 3.185. Find unique DNS entry: JSON response

```
{
  "dns_entry": {
    "domain": "domain1.example.org",
    "id": null,
    "ip": "192.168.1.1",
    "name": "instance1",
    "type": null
  }
}
```

Example 3.186. Find unique DNS entry: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<dns_entry ip="192.168.1.1" domain="domain1.example.org" type="None" id="None"
name="instance1"/>
```

This operation does not return a response body.

3.22.6. Delete DNS entry

Method	URI	Description
DELETE	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{name}	Deletes a specified DNS entry.

Normal response codes: 200

3.22.6.1. Request

This table shows the URI parameters for the delete dns entry request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.
{name}	String	DNS entry name assigned under a domain.

This operation does not require a request body.

3.22.7. List DNS entries

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ip-dns/{domain}/entries/{ip}	Lists DNS entries for a specified domain and IP.

Normal response codes: 200

3.22.7.1. Request

This table shows the URI parameters for the list dns entries request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{domain}	String	Registered DNS domain published by the DNS drivers.

This operation does not require a request body.

3.22.7.2. Response

Example 3.187. List DNS entries: JSON response

```
{
  "dns_entries": [
    {
      "domain": "domain1.example.org",
      "id": null,
      "ip": "192.168.1.1",
      "name": "instance1",
      "type": null
    }
  ]
}
```

Example 3.188. List DNS entries: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<dns_entries>
  <dns_entry ip="192.168.1.1" domain="domain1.example.org" type="None" id="None" name="instance1"/>
</dns_entries>
```

This operation does not return a response body.

3.23. Floating IP pools (os-floating-ip-pools)

Manage groups of floating IPs.

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ip-pools	Lists floating IP pools.

3.23.1. List floating IP pools

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ip-pools	Lists floating IP pools.

Normal response codes: 200

3.23.1.1. Request

This table shows the URI parameters for the list floating ip pools request:

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

This operation does not require a request body.

3.23.1.2. Response

Example 3.189. List floating IP pools: JSON response

```
{
  "floating_ip_pools": [
    {
      "name": "pool1"
    },
    {
      "name": "pool2"
    }
  ]
}
```

Example 3.190. List floating IP pools: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ip_pools>
  <floating_ip_pool name="pool1"/>
  <floating_ip_pool name="pool2"/>
</floating_ip_pools>
```

This operation does not return a response body.

3.24. Floating IPs (os-floating-ips)

Assign and allocate floating IP addresses to instances that run in an OpenStack cloud.

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ips	Lists floating IP addresses associated with the tenant or account.
POST	/v2/{tenant_id}/os-floating-ips	Allocates a new floating IP address to a tenant or account.
GET	/v2/{tenant_id}/os-floating-ips/{id}	Shows information for a specified floating IP address.

Method	URI	Description
DELETE	/v2/{tenant_id}/os-floating-ips/{id}	Deallocates the floating IP address associated with floating_IP_address_ID.
POST	/v2/{tenant_id}/servers/{server_id}/action	Adds a floating IP address to an instance.
POST	/v2/{tenant_id}/servers/{server_id}/action	Removes a floating IP from an instance.

3.24.1. List floating IPs

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ips	Lists floating IP addresses associated with the tenant or account.

Normal response codes: 200

3.24.1.1. Request

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

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

This operation does not require a request body.

3.24.1.2. Response

Example 3.191. List floating IPs: JSON response

```
{
  "floating_ips": [
    {
      "fixed_ip": null,
      "id": 1,
      "instance_id": null,
      "ip": "10.10.10.1",
      "pool": "nova"
    },
    {
      "fixed_ip": null,
      "id": 2,
      "instance_id": null,
      "ip": "10.10.10.2",
      "pool": "nova"
    }
  ]
}
```

Example 3.192. List floating IPs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ips>
  <floating_ip instance_id="None" ip="10.10.10.1" fixed_ip="None" id="1" pool="nova"/>
  <floating_ip instance_id="None" ip="10.10.10.2" fixed_ip="None" id="2" pool="nova"/>
</floating_ips>
```

This operation does not return a response body.

3.24.2. Allocate floating IP

Method	URI	Description
POST	/v2/{tenant_id}/os-floating-ips	Allocates a new floating IP address to a tenant or account.

This table shows the possible response codes for this operation:

Response Code	Name	Description
200		
400		If there are no floating IPs available, the extension returns an error code 400 with a message indicating that no more floating IPs are available.

3.24.2.1. Request

This table shows the URI parameters for the allocate floating ip request:

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

Example 3.193. Allocate floating IP: JSON request

```
{
  "pool": "nova"
}
```

Example 3.194. Allocate floating IP: XML request

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

This operation does not require a request body.

3.24.2.2. Response

Example 3.195. Allocate floating IP: JSON response

```
{
  "floating_ip": {
    "fixed_ip": null,
    "id": 1,
    "instance_id": null,
    "ip": "10.10.10.1",
    "pool": "nova"
  }
}
```

Example 3.196. Allocate floating IP: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ip instance_id="None" ip="10.10.10.1" fixed_ip="None" id="1" pool="nova"/>
```


This operation does not return a response body.

3.24.3. Show floating IP information

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ips/{id}	Shows information for a specified floating IP address.

Normal response codes: 200

3.24.3.1. Request

This table shows the URI parameters for the show floating ip information request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	String	The unique identifier associated with allocated floating IP address.

This operation does not require a request body.

3.24.3.2. Response

Example 3.197. Show floating IP information: JSON response

```
{
  "floating_ip": {
    "fixed_ip": null,
    "id": 1,
    "instance_id": null,
    "ip": "10.10.10.1",
    "pool": "nova"
  }
}
```

Example 3.198. Show floating IP information: XML response

```
<floating_ip instance_id="None" ip="10.10.10.1" fixed_ip="None" id="1" pool="nova"/>
```

This operation does not return a response body.

3.24.4. Deallocate floating IP

Method	URI	Description
DELETE	/v2/{tenant_id}/os-floating-ips/{id}	Deallocates the floating IP address associated with floating_IP_address_ID.

Normal response codes: 202

3.24.4.1. Request

This table shows the URI parameters for the deallocate floating ip request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	String	The unique identifier associated with allocated floating IP address.

This operation does not require a request body.

3.24.5. Add floating IP

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

Normal response codes: 202

3.24.5.1. Request

This table shows the URI parameters for the add floating ip request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{server_id}	UUID	The UUID for the server of interest to you.

Example 3.199. Add floating IP: JSON request

```
{
  "addFloatingIp": {
    "address": "10.10.10.1"
  }
}
```

Example 3.200. Add floating IP: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<addFloatingIp>
  <address>10.10.10.1</address>
</addFloatingIp>
```

This operation does not require a request body.

3.24.6. Remove floating IP

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Removes a floating IP from an instance.

Normal response codes: 202

3.24.6.1. Request

This table shows the URI parameters for the remove floating ip request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{server_id}	UUID	The UUID for the server of interest to you.

Example 3.201. Remove floating IP: JSON request

```
{
  "removeFloatingIp": {
    "address": "10.10.10.1"
  }
}
```

Example 3.202. Remove floating IP: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<removeFloatingIp>
  <address>10.10.10.1</address>
</removeFloatingIp>
```

This operation does not require a request body.

3.25. Floating IPs bulk (os-floating-ips-bulk)

Bulk create, delete, and list floating IPs. By default, the pool is named nova. Use the `os-floating-ip-pools` extension to view available pools.

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ips-bulk	Lists all floating IPs.
POST	/v2/{tenant_id}/os-floating-ips-bulk	Bulk-creates floating IPs.
POST	/v2/{tenant_id}/os-floating-ips-bulk/delete	Bulk-deletes floating IPs.
GET	/v2/{tenant_id}/os-floating-ips-bulk/{host_name}	Lists all floating IPs for a specified host.

3.25.1. List floating IPs

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ips-bulk	Lists all floating IPs.

Normal response codes: 200

3.25.1.1. Request

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

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

This operation does not require a request body.

3.25.1.2. Response

Example 3.203. List floating IPs: JSON response

```
{
  "floating_ip_info": [
    {
      "address": "10.10.10.1",
      "instance_uuid": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    },
    {
      "address": "10.10.10.2",
      "instance_uuid": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    },
    {
      "address": "10.10.10.3",
      "instance_uuid": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    }
  ]
}
```

Example 3.204. List floating IPs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ip_info>
  <item>
    <interface>eth0</interface>
    <instance_uuid>None</instance_uuid>
    <project_id>None</project_id>
```

```
<pool>nova</pool>
  <address>10.10.10.1</address>
</item>
<item>
  <interface>eth0</interface>
  <instance_uuid>None</instance_uuid>
  <project_id>None</project_id>
  <pool>nova</pool>
  <address>10.10.10.2</address>
</item>
<item>
  <interface>eth0</interface>
  <instance_uuid>None</instance_uuid>
  <project_id>None</project_id>
  <pool>nova</pool>
  <address>10.10.10.3</address>
</item>
</floating_ip_info>
```

This operation does not return a response body.

3.25.2. Create floating IPs

Method	URI	Description
POST	/v2/{tenant_id}/os-floating-ips-bulk	Bulk-creates floating IPs.

Normal response codes: 200

3.25.2.1. Request

This table shows the URI parameters for the create floating ips request:

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

Example 3.205. Create floating IPs: JSON request

```
{
  "floating_ips_bulk_create": {
    "ip_range": "192.168.1.0/24",
    "pool": "nova",
    "interface": "eth0"
  }
}
```

Example 3.206. Create floating IPs: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ips_bulk_create>
<ip_range>192.168.1.0/24</ip_range>
<pool>nova</pool>
<interface>eth0</interface>
</floating_ips_bulk_create>
```

This operation does not require a request body.

3.25.2.2. Response

Example 3.207. Create floating IPs: JSON response

```
{
  "floating_ips_bulk_create": {
    "interface": "eth0",
    "ip_range": "192.168.1.0/24",
    "pool": "nova"
  }
}
```

Example 3.208. Create floating IPs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ips_bulk_create>
  <interface>eth0</interface>
  <ip_range>192.168.1.0/24</ip_range>
```



```
<pool>nova</pool>  
</floating_ips_bulk_create>
```

This operation does not return a response body.

3.25.3. Bulk-delete floating IPs

Method	URI	Description
POST	/v2/{tenant_id}/os-floating-ips-bulk/delete	Bulk-deletes floating IPs.

Normal response codes: 200

3.25.3.1. Request

This table shows the URI parameters for the bulk-delete floating ips request:

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

Example 3.209. Bulk-delete floating IPs: JSON request

```
{
  "ip_range": "192.168.1.0/24"
}
```

Example 3.210. Bulk-delete floating IPs: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<ip_range>192.168.1.0/24</ip_range>
```

This operation does not require a request body.

3.25.3.2. Response

Example 3.211. Bulk-delete floating IPs: JSON response

```
{
  "floating_ips_bulk_delete": "192.168.1.0/24"
}
```

Example 3.212. Bulk-delete floating IPs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ips_bulk_delete>192.168.1.0/24</floating_ips_bulk_delete>
```

This operation does not return a response body.

3.25.4. List floating IPs by host

Method	URI	Description
GET	/v2/{tenant_id}/os-floating-ips-bulk/{host_name}	Lists all floating IPs for a specified host.

Normal response codes: 200

3.25.4.1. Request

This table shows the URI parameters for the list floating ips by host request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{host_name}	String	The name of the host of interest to you.

This operation does not require a request body.

3.25.4.2. Response

Example 3.213. List floating IPs by host: JSON response

```
{
  "floating_ip_info": [
    {
      "address": "10.10.10.3",
      "instance_uuid": null,
      "interface": "eth0",
      "pool": "nova",
      "project_id": null
    }
  ]
}
```

Example 3.214. List floating IPs by host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<floating_ip_info>
  <item>
    <interface>eth0</interface>
    <instance_uuid>None</instance_uuid>
    <project_id>None</project_id>
    <pool>nova</pool>
    <address>10.10.10.3</address>
  </item>
</floating_ip_info>
```

This operation does not return a response body.

3.26. Hosts (os-hosts)

Manage physical hosts.

Method	URI	Description
GET	/v2/{tenant_id}/os-hosts{?service, zone}	Lists hosts.
GET	/v2/{tenant_id}/os-hosts/{host_name}	Shows information for a specified host.
PUT	/v2/{tenant_id}/os-hosts/{host_name}	Enables a host or puts it in maintenance mode.
GET	/v2/{tenant_id}/os-hosts/{host_name}/startup	Starts a host.
GET	/v2/{tenant_id}/os-hosts/{host_name}/shutdown	Shuts down a host.
GET	/v2/{tenant_id}/os-hosts/{host_name}/reboot	Reboots a host.

3.26.1. List hosts

Method	URI	Description
GET	/v2/{tenant_id}/os-hosts{?service, zone}	Lists hosts.

Normal response codes: 200

3.26.1.1. Request

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

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

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

Name	Type	Description
service	String <i>(Optional)</i>	The service name. To filter the returned list by service name, specify ?service={service} in the URI.
zone	String <i>(Optional)</i>	The zone. To filter the returned list by zone, specify ?zone={zone} in the URI.

3.26.1.2. Response

Example 3.215. List hosts: JSON response

```
{
  "hosts": [
    {
      "host_name": "787f4f6dda1b409bb8b2f9082349690e",
      "service": "compute",
      "zone": "nova"
    },
    {
      "host_name": "a98b433151084aee8b1a986e28823b36",
      "service": "cert",
      "zone": "internal"
    },
    {
      "host_name": "c56158d13a884a87abf9171efb7de9d8",
      "service": "network",
      "zone": "internal"
    },
    {
      "host_name": "81d5cdcda0014918b3ebd3503a2e5c9a",
      "service": "scheduler",
      "zone": "internal"
    },
    {
      "host_name": "6e48bfe1a3304b7b86154326328750ae",
      "service": "conductor",
      "zone": "internal"
    }
  ]
}
```

```
    },
    {
      "host_name": "39f55087a1024d1380755951c945ca69",
      "service": "cells",
      "zone": "internal"
    },
    {
      "host_name": "7a9a6cb4701f4dee9048fe0bc25d0ee5",
      "service": "consoleauth",
      "zone": "internal"
    }
  ]
}
```

Example 3.216. List hosts: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hosts>
  <host zone="internal" host_name="02ff30050a0d4ce486e60e946a97d28f" service=
"conductor"/>
  <host zone="nova" host_name="69dd365b9eae459883a50d1fe7b3c40b" service=
"compute"/>
  <host zone="internal" host_name="059ca4da76cf49e8a3d08d2303542157" service=
"cert"/>
  <host zone="internal" host_name="ea70b9f0c530497fba8571deb4835ab0" service=
"network"/>
  <host zone="internal" host_name="7d52ad3988504fb18b0cdfd94dbd267b" service=
"scheduler"/>
  <host zone="internal" host_name="dc36bce962cc49c2ab916287bcf6c9d3" service=
"cells"/>
  <host zone="internal" host_name="4bd7bbcc80064d21a5fd0bd53318c6fa" service=
"consoleauth"/>
</hosts>
```

This operation does not return a response body.

3.26.2. Show host information

Method	URI	Description
GET	/v2/{tenant_id}/os-hosts/{host_name}	Shows information for a specified host.

Normal response codes: 200

3.26.2.1. Request

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

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

This operation does not require a request body.

3.26.2.2. Response

Example 3.217. Show host information: JSON response

```
{
  "host": [
    {
      "resource": {
        "cpu": 1,
        "disk_gb": 1028,
        "host": "5ca60c6792a1442f9471ff575443f94d",
        "memory_mb": 8192,
        "project": "(total)"
      }
    },
    {
      "resource": {
        "cpu": 0,
        "disk_gb": 0,
        "host": "5ca60c6792a1442f9471ff575443f94d",
        "memory_mb": 512,
        "project": "(used_now)"
      }
    },
    {
      "resource": {
        "cpu": 0,
        "disk_gb": 0,
        "host": "5ca60c6792a1442f9471ff575443f94d",
        "memory_mb": 0,
        "project": "(used_max)"
      }
    }
  ]
}
```

Example 3.218. Show host information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host>
  <resource>
    <project>(total)</project>
    <memory_mb>8192</memory_mb>
    <host>ecf3458ac6bf4a299cc2e0efa740f426</host>
    <cpu>1</cpu>
    <disk_gb>1028</disk_gb>
  </resource>
  <resource>
    <project>(used_now)</project>
    <memory_mb>512</memory_mb>
    <host>ecf3458ac6bf4a299cc2e0efa740f426</host>
    <cpu>0</cpu>
    <disk_gb>0</disk_gb>
  </resource>
  <resource>
    <project>(used_max)</project>
    <memory_mb>0</memory_mb>
    <host>ecf3458ac6bf4a299cc2e0efa740f426</host>
    <cpu>0</cpu>
    <disk_gb>0</disk_gb>
  </resource>
</host>
```

This operation does not return a response body.

3.26.3. Update host

Method	URI	Description
PUT	/v2/{tenant_id}/os-hosts/{host_name}	Enables a host or puts it in maintenance mode.

Normal response codes: 200

3.26.3.1. Request

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

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

Example 3.219. Update host: JSON request

```
{
  "status": "enable",
  "maintenance_mode": "disable"
}
```

Example 3.220. Update host: XML request

```
<?xml version="1.0" encoding="UTF-8" ?>
<updates>
  <status>enable</status>
  <maintenance_mode>disable</maintenance_mode>
</updates>
```

This operation does not require a request body.

3.26.3.2. Response

Example 3.221. Update host: JSON response

```
{
  "host": "0738dca90a8c43fdadd0be28715520e2",
  "maintenance_mode": "off_maintenance",
  "status": "enabled"
}
```

Example 3.222. Update host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host status="enabled" maintenance_mode="off_maintenance" host=
"d85f05519b57457c83da18c39fa8e00d" />
```

This operation does not return a response body.

3.26.4. Start host

Method	URI	Description
GET	/v2/{tenant_id}/os-hosts/{host_name}/startup	Starts a host.

Normal response codes: 200

3.26.4.1. Request

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

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

This operation does not require a request body.

3.26.4.2. Response

Example 3.223. Start host: JSON response

```
{
  "host": "57f5de2fa5b44f14974a4f50b9ffc8",
  "power_action": "startup"
}
```

Example 3.224. Start host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host host="7fae168ed18140d5a785ade2ac1bd420" power_action="startup"/>
```

This operation does not return a response body.

3.26.5. Shut down host

Method	URI	Description
GET	/v2/{tenant_id}/os-hosts/{host_name}/shutdown	Shuts down a host.

Normal response codes: 200

3.26.5.1. Request

This table shows the URI parameters for the shut down host request:

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

This operation does not require a request body.

3.26.5.2. Response

Example 3.225. Shut down host: JSON response

```
{
  "host": "d2576862a2144ee6ad37d9e1938460a2",
  "power_action": "shutdown"
}
```

Example 3.226. Shut down host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host host="c82ca6da579440ac930ddee0e6530176" power_action="shutdown"/>
```

This operation does not return a response body.

3.26.6. Reboot host

Method	URI	Description
GET	/v2/{tenant_id}/os-hosts/{host_name}/reboot	Reboots a host.

Normal response codes: 200

3.26.6.1. Request

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

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

This operation does not require a request body.

3.26.6.2. Response

Example 3.227. Reboot host: JSON response

```
{
  "host": "066bf157ab50481d8c607cfe584b2230",
  "power_action": "reboot"
}
```

Example 3.228. Reboot host: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<host host="ce8c5f8cde4a46ffb01dec0788ad4dfc" power_action="reboot"/>
```

This operation does not return a response body.

3.27. Hypervisors (os-hypervisors)

Display extra statistical information from the machine that hosts the hypervisor through the API for the hypervisor (XenAPI or KVM/libvirt).

Method	URI	Description
GET	/v2/{tenant_id}/os-hypervisors	Lists hypervisors information for each server obtained through the hypervisor-specific API, such as libvirt or XenAPI.
GET	/v2/{tenant_id}/os-hypervisors/detail	Shows information for a specified hypervisor. Typically configured as an admin-only extension by using <code>policy.json</code> settings.
GET	/v2/{tenant_id}/os-hypervisors/statistics	Shows hypervisor statistics over all compute nodes.
GET	/v2/{tenant_id}/os-hypervisors/{hypervisor_hostname}	Shows the up time for a specified hypervisor.
GET	/v2/{tenant_id}/os-hypervisors/{hypervisor_hostname}/servers	Lists instances that belong to specific hypervisors.

3.27.1. List hypervisors

Method	URI	Description
GET	/v2/{tenant_id}/os-hypervisors	Lists hypervisors information for each server obtained through the hypervisor-specific API, such as libvirt or XenAPI.

Normal response codes: 200

3.27.1.1. Request

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

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

This operation does not require a request body.

3.27.1.2. Response

Example 3.229. List hypervisors: JSON response

```
{
  "hypervisors": [
    {
      "hypervisor_hostname": "fake-mini",
      "id": 1
    }
  ]
}
```

Example 3.230. List hypervisors: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisors>
  <hypervisor id="1" hypervisor_hostname="fake-mini"/>
</hypervisors>
```

This operation does not return a response body.

3.27.2. Show hypervisor information

Method	URI	Description
GET	/v2/{tenant_id}/os-hypervisors/detail	Shows information for a specified hypervisor. Typically configured as an admin-only extension by using <code>policy.json</code> settings.

Normal response codes: 200

3.27.2.1. Request

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

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

This operation does not require a request body.

3.27.2.2. Response

Example 3.231. Show hypervisor information: JSON response

```
{
  "hypervisors": [
    {
      "cpu_info": "?",
      "current_workload": 0,
      "disk_available_least": null,
      "free_disk_gb": 1028,
      "free_ram_mb": 7680,
      "hypervisor_hostname": "fake-mini",
      "hypervisor_type": "fake",
      "hypervisor_version": 1,
      "id": 1,
      "local_gb": 1028,
      "local_gb_used": 0,
      "memory_mb": 8192,
      "memory_mb_used": 512,
      "running_vms": 0,
      "service": {
        "host": "1e0d7892083548cfb347e782d3b20342",
        "id": 2
      },
      "vcpus": 1,
      "vcpus_used": 0
    }
  ]
}
```

Example 3.232. Show hypervisor information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisors>
  <hypervisor vcpus_used="0" hypervisor_type="fake" local_gb_used="0"
hypervisor_hostname="fake-mini" memory_mb_used="512" memory_mb="8192"
```

```
current_workload="0" vcpus="1" cpu_info="?" running_vms="0" free_disk_gb="1028" hypervisor_version="1" disk_available_least="None" local_gb="1028" free_ram_mb="7680" id="1">  
  <service host="4400f556a66d44ce95dfa61e75a23aaf" id="2"/>  
</hypervisor>  
</hypervisors>
```

This operation does not return a response body.

3.27.3. Show statistics for hypervisors

Method	URI	Description
GET	/v2/{tenant_id}/os-hypervisors/statistics	Shows hypervisor statistics over all compute nodes.

Normal response codes: 200

3.27.3.1. Request

This table shows the URI parameters for the show statistics for hypervisors request:

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

This operation does not require a request body.

3.27.3.2. Response

Example 3.233. Show statistics for hypervisors: JSON response

```
{
  "hypervisor_statistics": {
    "count": 1,
    "current_workload": 0,
    "disk_available_least": 0,
    "free_disk_gb": 1028,
    "free_ram_mb": 7680,
    "local_gb": 1028,
    "local_gb_used": 0,
    "memory_mb": 8192,
    "memory_mb_used": 512,
    "running_vms": 0,
    "vcpus": 1,
    "vcpus_used": 0
  }
}
```

Example 3.234. Show statistics for hypervisors: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisor_statistics count="1" vcpus_used="0" local_gb_used="0" memory_mb=
"8192" current_workload="0" vcpus="1" running_vms="0" free_disk_gb="1028"
disk_available_least="0" local_gb="1028" free_ram_mb="7680" memory_mb_used=
"512"/>
```

This operation does not return a response body.

3.27.4. Show hypervisor up time

Method	URI	Description
GET	/v2/{tenant_id}/os-hypervisors/{hypervisor_hostname}	Shows the up time for a specified hypervisor.

Normal response codes: 200

3.27.4.1. Request

This table shows the URI parameters for the show hypervisor up time request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{hypervisor_hostname}	String	The name of the host that runs the hypervisor.

This operation does not require a request body.

3.27.4.2. Response

Example 3.235. Show hypervisor up time: JSON response

```
{
  "hypervisor": {
    "hypervisor_hostname": "fake-mini",
    "id": 1,
    "uptime": " 08:32:11 up 93 days, 18:25, 12 users,  load average: 0.20,
0.12, 0.14"
  }
}
```

Example 3.236. Show hypervisor up time: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisor uptime=" 08:32:11 up 93 days, 18:25, 12 users,  load average: 0.
20, 0.12, 0.14" id="1" hypervisor_hostname="fake-mini"/>
```

This operation does not return a response body.

3.27.5. List instances for hypervisors

Method	URI	Description
GET	/v2/{tenant_id}/os-hypervisors/{hypervisor_hostname}/servers	Lists instances that belong to specific hypervisors.

Normal response codes: 200

3.27.5.1. Request

This table shows the URI parameters for the list instances for hypervisors request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{hypervisor_hostname}	String	The name of the host that runs the hypervisor.

This operation does not require a request body.

3.27.5.2. Response

Example 3.237. List instances for hypervisors: JSON response

```
{
  "hypervisors": [
    {
      "hypervisor_hostname": "fake-mini",
      "id": 1
    }
  ]
}
```

Example 3.238. List instances for hypervisors: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<hypervisors>
  <hypervisor id="1" hypervisor_hostname="fake-mini">
    <servers/>
  </hypervisor>
</hypervisors>
```

This operation does not return a response body.

3.28. Server actions (os-instance-actions)

List available actions for a specified server. Administrators can get details for a specified action for a specified server.

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-instance-actions	Lists available actions for a specified server. Deployers set permissions for this request in the <code>policy.json</code> file. By default, all users can list actions.

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-instance-actions/{action_id}	Gets details for a specified action for a specified server instance. Deployers set permissions for this request in the <code>policy.json</code> file. By default, only administrators can get details for an action.

3.28.1. List server actions

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-instance-actions	Lists available actions for a specified server. Deployers set permissions for this request in the <code>policy.json</code> file. By default, all users can list actions.

Normal response codes: 200

3.28.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server instance for which you want to list actions.

This operation does not require a request body.

3.28.1.2. Response

Example 3.239. List server actions: JSON response

```
{
  "instanceActions": [
    {
      "action": "resize",
      "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
      "message": "",
      "project_id": "842",
      "request_id": "req-25517360-b757-47d3-be45-0e8d2a01b36a",
      "start_time": "2012-12-05 01:00:00.000000",
      "user_id": "789"
    },
    {
      "action": "reboot",
      "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
      "message": "",
      "project_id": "147",
      "request_id": "req-3293a3f1-b44c-4609-b8d2-d81b105636b8",
      "start_time": "2012-12-05 00:00:00.000000",
      "user_id": "789"
    }
  ]
}
```

Example 3.240. List server actions: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<instanceActions>
  <instanceAction instance_uuid="b48316c5-71e8-45e4-9884-6c78055b9b13"
    user_id="789" start_time="2012-12-05 01:00:00.000000" request_id=
"req-25517360-b757-47d3-be45-0e8d2a01b36a" action="resize" message=""
    project_id="842"/>
</instanceActions>
```

```
<instanceAction instance_uuid="b48316c5-71e8-45e4-9884-6c78055b9b13"  
  user_id="789" start_time="2012-12-05 00:00:00.000000" request_id=  
  "req-3293a3f1-b44c-4609-b8d2-d81b105636b8" action="reboot" message=""  
  project_id="147" />  
</instanceActions>
```

This operation does not return a response body.

3.28.2. Get action details

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-instance-actions/{action_id}	Gets details for a specified action for a specified server instance. Deployers set permissions for this request in the <code>policy.json</code> file. By default, only administrators can get details for an action.

Normal response codes: 200, 203

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

3.28.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID of the server instance for which you want to list actions.
{action_id}	UUID	The UUID of the action for which you want to get details.

This operation does not require a request body.

3.28.2.2. Response

Example 3.241. Get action details: JSON response

```
{
  "instanceAction": {
    "action": "reboot",
    "events": [
      {
        "event": "schedule",
        "finish_time": "2012-12-05 01:02:00.000000",
        "result": "Success",
        "start_time": "2012-12-05 01:00:02.000000",
        "traceback": ""
      },
      {
        "event": "compute_create",
        "finish_time": "2012-12-05 01:04:00.000000",
        "result": "Success",
        "start_time": "2012-12-05 01:03:00.000000",
        "traceback": ""
      }
    ]
  },
  "instance_uuid": "b48316c5-71e8-45e4-9884-6c78055b9b13",
  "message": "",
  "project_id": "147",
  "request_id": "req-3293a3f1-b44c-4609-b8d2-d81b105636b8",
  "start_time": "2012-12-05 00:00:00.000000",
  "user_id": "789"
}
```

```
}  
}
```

Example 3.242. Get action details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<instanceAction instance_uuid="b48316c5-71e8-45e4-9884-6c78055b9b13" user_id="789" start_time="2012-12-05T00:00:00.000000" request_id="req-3293a3f1-b44c-4609-b8d2-d81b105636b8" action="reboot" message="" project_id="147">  
  <events finish_time="2012-12-05 01:02:00.000000" start_time="2012-12-05 01:00:02.000000" traceback="" event="schedule" result="Success"/>  
  <events finish_time="2012-12-05 01:04:00.000000" start_time="2012-12-05 01:03:00.000000" traceback="" event="compute_create" result="Success"/>  
</instanceAction>
```

This operation does not return a response body.

3.29. Keypairs (os-keypairs)

Generate, import, and delete SSH keys.

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

3.29.1. List keypairs

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

Normal response codes: 200

3.29.1.1. Request

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

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

This operation does not require a request body.

3.29.1.2. Response

Example 3.243. List keypairs: JSON response

```
{
  "keypairs": [
    {
      "keypair": {
        "fingerprint":
"15:b0:f8:b3:f9:48:63:71:cf:7b:5b:38:6d:44:2d:4a",
        "name": "keypair-601a2305-4f25-41ed-89c6-2a966fc8027a",
        "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC
+Eo/RZRngaGTkFs7I62ZjsIlo79KklKbMXi8F+KITD4bVQHhN+kV
+4gRgkgCRbdoDqoGfpaDFs877DYX9n4z6FrAIZ4PES8TNKhatifpn9NdQYWA
+IkU8CuvlEKGuFpKri/k7JLos/gHi2hy7QUwgtRvcefvD/vgQZOVw/mGR9Q== Generated by
Nova\n"
      }
    }
  ]
}
```

Example 3.244. List keypairs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<keypairs>
  <keypair>
    <public_key>ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGCWdUoGD7qz9kjbLoY2L0S5CdhUS8RvQ1g62OTgvmWE/
bEKDLwaTIFEEpN/0huGk/nxvVZ6VOhvlESKC3o9dZ2NDk0C4sBsrvJ41uWdlhbq72sDGzVEkJ
+925CraioSAbMprK5Ea7UPWbR8laqrY1TsKtcuxiGJ936bOPIXW12h6Q== Generated by Nova
</public_key>
    <name>keypair-a4c7d228-218b-4c4c-9d99-62e7878ebblb</name>
    <fingerprint>62:32:23:67:56:ee:6f:51:4c:03:ce:b8:00:f9:41:ff</fingerprint>
  </keypair>
</keypairs>
```

This operation does not return a response body.

3.29.2. Create or import keypair

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

Normal response codes: 200

3.29.2.1. Request

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

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

Example 3.245. Create or import keypair: XML request

```
<keypair>
  <name>keypair-96bbe50e-05e1-4d59-9115-4779a3ebcc2e</name>
  <public_key>ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDx8nkQv/zgGgB4rMYmIf
+6A4l6Rr+o/6lHBQdW5aYd44bd8JttDCE/F/pNRr0lRE+PiqSPO8nDPHw0010JeMH9gYggnFlyY3/
OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/0d3ac30bNH6Sw9JD9UZHYcpSxsIbECHw==
Generated by Nova</public_key>
</keypair>
```

Example 3.246. Create or import keypair: JSON request

```
{
  "keypair": {
    "name": "keypair-dab428fe-6186-4a14-b3de-92131f76cd39",
    "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDx8nkQv/
zgGgB4rMYmIf+6A4l6Rr+o/6lHBQdW5aYd44bd8JttDCE/F/pNRr0lRE
+PiqSPO8nDPHw0010JeMH9gYggnFlyY3/OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/
0d3ac30bNH6Sw9JD9UZHYcpSxsIbECHw== Generated by Nova"
  }
}
```

This operation does not require a request body.

3.29.2.2. Response

Example 3.247. Create or import keypair: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<keypair>
  <public_key>ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDx8nkQv/zgGgB4rMYmIf
+6A4l6Rr+o/6lHBQdW5aYd44bd8JttDCE/F/pNRr0lRE+PiqSPO8nDPHw0010JeMH9gYggnFlyY3/
OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXpD/0d3ac30bNH6Sw9JD9UZHYcpSxsIbECHw==
Generated by Nova</public_key>
  <user_id>fake</user_id>
  <name>keypair-96bbe50e-05e1-4d59-9115-4779a3ebcc2e</name>
  <fingerprint>1e:2c:9b:56:79:4b:45:77:f9:ca:7a:98:2c:b0:d5:3c</fingerprint>
</keypair>
```

Example 3.248. Create or import keypair: JSON response

```
{
```

```
"keypair": {
  "fingerprint": "1e:2c:9b:56:79:4b:45:77:f9:ca:7a:98:2c:b0:d5:3c",
  "name": "keypair-dab428fe-6186-4a14-b3de-92131f76cd39",
  "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDx8nkQv/
zgGgB4rMYmIf+6A416Rr+o/61HBQdW5ayd44bd8JttDCE/F/pNRr01RE
+PiqSP08nDPHw0010JeMH9gYgnnFlyY3/OcJ02RhIPyyxYpv9FhY+2YiUkpwFOcLImyrxEsYXPD/
0d3ac30bNH6Sw9JD9UZHYcpSxsIbECHw== Generated by Nova",
  "user_id": "fake"
}
```

This operation does not return a response body.

3.29.3. Delete keypair

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

Normal response codes: 202

3.29.3.1. Request

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

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

This operation does not require a request body.

3.29.4. Show keypair information

Method	URI	Description
GET	/v2/{tenant_id}/os-keypairs/{keypair_name}	Shows a keypair associated with the account.

Normal response codes: 200

3.29.4.1. Request

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

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

This operation does not require a request body.

3.29.4.2. Response

Example 3.249. Show keypair information: JSON response

```
{
  "keypair": {
    "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDCSLxfzqB
+e5yHdUSXvbxKkajjlfuhV
+GardseqPjfkKJ6no5echpin7dJp0FLXMJKxJZE3WWIRu25CQrJntmi7no27RkDfAGaTFbjz3DWY4A1HLeKAB5tFTwYQ
DY7UvKaatwLh4Wvh5vMmXgF7AFzdkI28urwo+Q== nova@use03147k5-eth0\n",
    "name": "hpdefault",
    "fingerprint": "8b:2f:aa:b0:b8:97:dc:c8:50:aa:d4:8e:d0:34:61:d9"
  }
}
```

3.30. Migrations (os-migrations)

Administrators only. Fetch in-progress migrations for a region or a specified cell in a region.

Method	URI	Description
GET	/v2/{tenant_id}/os-migrations{?host,status,cell_name}	Enables an administrative user to fetch in-progress migrations for a region or specified cell in a region.

3.30.1. Get migrations

Method	URI	Description
GET	/v2/{tenant_id}/os-migrations{?host,status,cell_name}	Enables an administrative user to fetch in-progress migrations for a region or specified cell in a region.

Normal response codes: 200

3.30.1.1. Request

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

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

This operation does not require a request body.

3.30.1.2. Response

Example 3.250. Get migrations: JSON response

```
{
  "migrations": [
    {
      "created_at": "2012-10-29T13:42:02.000000",
      "dest_compute": "compute2",
      "dest_host": "1.2.3.4",
      "dest_node": "node2",
      "id": 1234,
      "instance_uuid": "instance_id_123",
      "new_instance_type_id": 2,
      "old_instance_type_id": 1,
      "source_compute": "compute1",
      "source_node": "node1",
      "status": "Done",
      "updated_at": "2012-10-29T13:42:02.000000"
    },
    {
      "created_at": "2013-10-22T13:42:02.000000",
      "dest_compute": "compute20",
      "dest_host": "5.6.7.8",
      "dest_node": "node20",
      "id": 5678,
      "instance_uuid": "instance_id_456",
      "new_instance_type_id": 6,
      "old_instance_type_id": 5,
      "source_compute": "compute10",
      "source_node": "node10",
      "status": "Done",
      "updated_at": "2013-10-22T13:42:02.000000"
    }
  ]
}
```

Example 3.251. Get migrations: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<migrations>
  <migration dest_host="1.2.3.4" status="Done" old_instance_type_id="1"
updated_at="2012-10-29 13:42:02" dest_compute="compute2" created_at=
"2012-10-29 13:42:02" source_node="node1" instance_uuid="instance_id_123"
dest_node="node2" id="1234" new_instance_type_id="2" source_compute=
"compute1"/>
  <migration dest_host="5.6.7.8" status="Done" old_instance_type_id="5"
updated_at="2013-10-22 13:42:02" dest_compute="compute20" created_at=
"2013-10-22 13:42:02" source_node="node10" instance_uuid="instance_id_456"
dest_node="node20" id="5678" new_instance_type_id="6" source_compute=
"compute10"/>
</migrations>
```

This operation does not return a response body.

3.31. Networks (os-networks)

Show network information for or delete networks. Also, disassociate a network from a project if you use vlan mode.

Method	URI	Description
POST	/v2/{tenant_id}	Creates a network.
GET	/v2/{tenant_id}/os-networks	Lists networks that are available to the tenant.
POST	/v2/{tenant_id}/os-networks/add	Adds a specified network to a project.
GET	/v2/{tenant_id}/os-networks/{id}	Shows information for a specified network.
DELETE	/v2/{tenant_id}/os-networks/{id}	Deletes a specified network.
POST	/v2/{tenant_id}/os-networks/{id}/action	Associates a specified network with a host.
POST	/v2/{tenant_id}/os-networks/{id}/action	Disassociates the host from a specified network.
POST	/v2/{tenant_id}/os-networks/{id}/action	Disassociates a specified network from a project so that the network can be reused.
POST	/v2/{tenant_id}/os-networks/{id}/action	Disassociates the project from a specified network.

3.31.1. Create network

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

Normal response codes: 202

3.31.1.1. Request

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

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

Example 3.252. Create network: JSON request

```
{
  "network": {
    "label": "new net 111",
    "cidr": "10.20.105.0/24"
  }
}
```

Example 3.253. Create network: XML request

```
<network>
  <label>new net 111</label>
  <cidr>10.20.105.0/24</cidr>
</network>
```

This operation does not require a request body.

3.31.1.2. Response

Example 3.254. Create network: JSON response

```
{
  "network": {
    "bridge": null,
    "bridge_interface": null,
    "broadcast": "10.20.105.255",
    "cidr": "10.20.105.0/24",
    "cidr_v6": null,
    "created_at": null,
    "deleted": null,
    "deleted_at": null,
    "dhcp_start": "10.20.105.2",
    "dns1": null,
    "dns2": null,
    "gateway": "10.20.105.1",
    "gateway_v6": null,
    "host": null,
    "id": "668687f9-d724-4976-a6f4-a6fd3ad83da3",
    "injected": null,
    "label": "new net 111",
  }
}
```



```
    "multi_host": null,  
    "netmask": "255.255.255.0",  
    "netmask_v6": null,  
    "priority": null,  
    "project_id": null,  
    "rxtx_base": null,  
    "updated_at": null,  
    "vlan": null,  
    "vpn_private_address": null,  
    "vpn_public_address": null,  
    "vpn_public_port": null  
  }  
}
```

Example 3.255. Create network: XML response

```
<?xml version='1.0' encoding='UTF-8'?>  
<network>  
  <bridge>None</bridge>  
  <vpn_public_port>None</vpn_public_port>  
  <dhcp_start>10.20.105.2</dhcp_start>  
  <bridge_interface>None</bridge_interface>  
  <updated_at>None</updated_at>  
  <id>1bbbed2b-0daa-47a1-b869-1981c29150b1</id>  
  <cidr_v6>None</cidr_v6>  
  <deleted_at>None</deleted_at>  
  <gateway>10.20.105.1</gateway>  
  <rxtx_base>None</rxtx_base>  
  <label>new net 111</label>  
  <priority>None</priority>  
  <project_id>None</project_id>  
  <vpn_private_address>None</vpn_private_address>  
  <deleted>False</deleted>  
  <vlan>None</vlan>  
  <broadcast>10.20.105.255</broadcast>  
  <netmask>255.255.255.0</netmask>  
  <injected>None</injected>  
  <cidr>10.20.105.0/24</cidr>  
  <vpn_public_address>None</vpn_public_address>  
  <multi_host>None</multi_host>  
  <dns2>None</dns2>  
  <created_at>None</created_at>  
  <host>None</host>  
  <gateway_v6>None</gateway_v6>  
  <netmask_v6>None</netmask_v6>  
  <dns1>None</dns1>  
</network>
```

This operation does not return a response body.

3.31.2. List networks

Method	URI	Description
GET	/v2/{tenant_id}/os-networks	Lists networks that are available to the tenant.

Normal response codes: 200

3.31.2.1. Request

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

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

This operation does not require a request body.

3.31.2.2. Response

Example 3.256. List networks: JSON response

```
{
  "networks": [
    {
      "bridge": "br100",
      "bridge_interface": "eth0",
      "broadcast": "10.0.0.7",
      "cidr": "10.0.0.0/29",
      "cidr_v6": null,
      "created_at": "2011-08-15 06:19:19.387525",
      "deleted": false,
      "deleted_at": null,
      "dhcp_start": "10.0.0.3",
      "dns1": null,
      "dns2": null,
      "gateway": "10.0.0.1",
      "gateway_v6": null,
      "host": "nsokolov-desktop",
      "id": "20c8acc0-f747-4d71-a389-46d078ebf047",
      "injected": false,
      "label": "mynet_0",
      "multi_host": false,
      "netmask": "255.255.255.248",
      "netmask_v6": null,
      "priority": null,
      "project_id": "1234",
      "rxtx_base": null,
      "updated_at": "2011-08-16 09:26:13.048257",
      "vlan": 100,
      "vpn_private_address": "10.0.0.2",
      "vpn_public_address": "127.0.0.1",
      "vpn_public_port": 1000
    },
    {
      "bridge": "br101",
      "bridge_interface": "eth0",
```

```

        "broadcast": "10.0.0.15",
        "cidr": "10.0.0.10/29",
        "cidr_v6": null,
        "created_at": "2011-08-15 06:19:19.885495",
        "deleted": false,
        "deleted_at": null,
        "dhcp_start": "10.0.0.11",
        "dns1": null,
        "dns2": null,
        "gateway": "10.0.0.9",
        "gateway_v6": null,
        "host": null,
        "id": "20c8acc0-f747-4d71-a389-46d078ebf000",
        "injected": false,
        "label": "mynet_1",
        "multi_host": false,
        "netmask": "255.255.255.248",
        "netmask_v6": null,
        "priority": null,
        "project_id": null,
        "rxtx_base": null,
        "updated_at": null,
        "vlan": 101,
        "vpn_private_address": "10.0.0.10",
        "vpn_public_address": null,
        "vpn_public_port": 1001
    }
}
]
}

```

Example 3.257. List networks: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<networks>
  <network>
    <bridge>br100</bridge>
    <vpn_public_port>1000</vpn_public_port>
    <dhcp_start>10.0.0.3</dhcp_start>
    <bridge_interface>eth0</bridge_interface>
    <updated_at>2011-08-16 09:26:13.048257</updated_at>
    <id>20c8acc0-f747-4d71-a389-46d078ebf047</id>
    <cidr_v6>None</cidr_v6>
    <deleted_at>None</deleted_at>
    <gateway>10.0.0.1</gateway>
    <rxtx_base>None</rxtx_base>
    <label>mynet_0</label>
    <priority>None</priority>
    <project_id>1234</project_id>
    <vpn_private_address>10.0.0.2</vpn_private_address>
    <deleted>False</deleted>
    <vlan>100</vlan>
    <broadcast>10.0.0.7</broadcast>
    <netmask>255.255.255.248</netmask>
    <injected>False</injected>
    <cidr>10.0.0.0/29</cidr>
    <vpn_public_address>127.0.0.1</vpn_public_address>
    <multi_host>False</multi_host>
    <dns2>None</dns2>
    <created_at>2011-08-15 06:19:19.387525</created_at>
    <host>nsokolov-desktop</host>
  </network>
</networks>

```

```
<gateway_v6>None</gateway_v6>
<netmask_v6>None</netmask_v6>
<dns1>None</dns1>
</network>
<network>
  <bridge>br101</bridge>
  <vpn_public_port>1001</vpn_public_port>
  <dhcp_start>10.0.0.11</dhcp_start>
  <bridge_interface>eth0</bridge_interface>
  <updated_at>None</updated_at>
  <id>20c8acc0-f747-4d71-a389-46d078ebf000</id>
  <cidr_v6>None</cidr_v6>
  <deleted_at>None</deleted_at>
  <gateway>10.0.0.9</gateway>
  <rxtx_base>None</rxtx_base>
  <label>mynet_1</label>
  <priority>None</priority>
  <project_id>None</project_id>
  <vpn_private_address>10.0.0.10</vpn_private_address>
  <deleted>False</deleted>
  <vlan>101</vlan>
  <broadcast>10.0.0.15</broadcast>
  <netmask>255.255.255.248</netmask>
  <injected>False</injected>
  <cidr>10.0.0.10/29</cidr>
  <vpn_public_address>None</vpn_public_address>
  <multi_host>False</multi_host>
  <dns2>None</dns2>
  <created_at>2011-08-15 06:19:19.885495</created_at>
  <host>None</host>
  <gateway_v6>None</gateway_v6>
  <netmask_v6>None</netmask_v6>
  <dns1>None</dns1>
</network>
</networks>
```

This operation does not return a response body.

3.31.3. Add network

Method	URI	Description
POST	/v2/{tenant_id}/os-networks/add	Adds a specified network to a project.

Normal response codes: 202

3.31.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	Uuid	The UUID of the network to add to the project. Specify <code>None</code> to choose a random available network.

Example 3.258. Add network: JSON request

```
{"id": "1"}
```

Example 3.259. Add network: XML request

```
<id>1</id>
```

This operation does not require a request body.

3.31.4. Show network information

Method	URI	Description
GET	/v2/{tenant_id}/os-networks/{id}	Shows information for a specified network.

Normal response codes: 200

3.31.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

This operation does not require a request body.

3.31.4.2. Response

Example 3.260. Show network information: JSON response

```
{
  "network": {
    "bridge": "br100",
    "bridge_interface": "eth0",
    "broadcast": "10.0.0.7",
    "cidr": "10.0.0.0/29",
    "cidr_v6": null,
    "created_at": "2011-08-15 06:19:19.387525",
    "deleted": false,
    "deleted_at": null,
    "dhcp_start": "10.0.0.3",
    "dns1": null,
    "dns2": null,
    "gateway": "10.0.0.1",
    "gateway_v6": null,
    "host": "nsokolov-desktop",
    "id": "20c8acc0-f747-4d71-a389-46d078ebf047",
    "injected": false,
    "label": "mynet_0",
    "multi_host": false,
    "netmask": "255.255.255.248",
    "netmask_v6": null,
    "priority": null,
    "project_id": "1234",
    "rxtx_base": null,
    "updated_at": "2011-08-16 09:26:13.048257",
    "vlan": 100,
    "vpn_private_address": "10.0.0.2",
    "vpn_public_address": "127.0.0.1",
    "vpn_public_port": 1000
  }
}
```

Example 3.261. Show network information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<network>
  <bridge>br100</bridge>
  <vpn_public_port>1000</vpn_public_port>
  <dhcp_start>10.0.0.3</dhcp_start>
  <bridge_interface>eth0</bridge_interface>
  <updated_at>2011-08-16 09:26:13.048257</updated_at>
  <id>20c8acc0-f747-4d71-a389-46d078ebf047</id>
  <cidr_v6>None</cidr_v6>
  <deleted_at>None</deleted_at>
  <gateway>10.0.0.1</gateway>
  <rxtx_base>None</rxtx_base>
  <label>mynet_0</label>
  <priority>None</priority>
  <project_id>1234</project_id>
  <vpn_private_address>10.0.0.2</vpn_private_address>
  <deleted>False</deleted>
  <vlan>100</vlan>
  <broadcast>10.0.0.7</broadcast>
  <netmask>255.255.255.248</netmask>
  <injected>False</injected>
  <cidr>10.0.0.0/29</cidr>
  <vpn_public_address>127.0.0.1</vpn_public_address>
  <multi_host>False</multi_host>
  <dns2>None</dns2>
  <created_at>2011-08-15 06:19:19.387525</created_at>
  <host>nsokolov-desktop</host>
  <gateway_v6>None</gateway_v6>
  <netmask_v6>None</netmask_v6>
  <dns1>None</dns1>
</network>
```

This operation does not return a response body.

3.31.5. Delete network

Method	URI	Description
DELETE	/v2/{tenant_id}/os-networks/{id}	Deletes a specified network.

Normal response codes: 202

3.31.5.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

This operation does not require a request body.

3.31.6. Associate host

Method	URI	Description
POST	/v2/{tenant_id}/os-networks/{id}/action	Associates a specified network with a host.

Normal response codes: 202

3.31.6.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

Example 3.262. Associate host: JSON request

```
{
  "associate_host": "testHost"
}
```

Example 3.263. Associate host: XML request

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

This operation does not require a request body.

3.31.7. Disassociate host

Method	URI	Description
POST	/v2/{tenant_id}/os-networks/{id}/action	Disassociates the host from a specified network.

Normal response codes: 202

3.31.7.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

Example 3.264. Disassociate host: JSON request

```
{
  "disassociate_host": null
}
```

Example 3.265. Disassociate host: XML request

```
<disassociate_host/>
```

This operation does not require a request body.

3.31.8. Disassociate network

Method	URI	Description
POST	/v2/{tenant_id}/os-networks/{id}/action	Disassociates a specified network from a project so that the network can be reused.

Normal response codes: 202

3.31.8.1. Request

This table shows the URI parameters for the disassociate network request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

Example 3.266. Disassociate network: JSON request

```
{
  "disassociate": null
}
```

Example 3.267. Disassociate network: XML request

```
<disassociate/>
```

This operation does not require a request body.

3.31.9. Disassociate project

Method	URI	Description
POST	/v2/{tenant_id}/os-networks/{id}/action	Disassociates the project from a specified network.

Normal response codes: 202

3.31.9.1. Request

This table shows the URI parameters for the disassociate project request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{id}	UUID	The unique identifier associated with the network.

Example 3.268. Disassociate project: JSON request

```
{
  "disassociate_project": null
}
```

Example 3.269. Disassociate project: XML request

```
<disassociate_project/>
```

This operation does not require a request body.

3.32. Quota sets (os-quota-sets)

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

Method	URI	Description
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Shows quotas for a tenant.
PUT	/v2/{tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.
GET	/v2/{tenant_id}/os-quota-sets/defaults	Gets default quotas for a tenant.
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a specified tenant and user.
POST	/v2/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a specified tenant/project and user.
GET	/v2/{tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}	Shows details for quotas for a specified tenant and user.

3.32.1. Show quotas

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

Normal response codes: 200

3.32.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project in a multi-tenancy cloud.
{tenant_id}	String	The ID for the tenant for which you want to show quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the admin tenant.

This operation does not require a request body.

3.32.1.2. Response

Example 3.270. Show quotas response: JSON

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

Example 3.271. Show quotas response: XML

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

```
<ram>51200</ram>  
<security_group_rules>20</security_group_rules>  
<security_groups>10</security_groups>  
</quota_set>
```

This operation does not return a response body.

3.32.2. Update quotas

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

Normal response codes: 200

3.32.2.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or project in a multi-tenancy cloud.
{tenant_id}	String	The ID for the tenant for which you want to show quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the admin tenant.

Example 3.272. Update quotas response: JSON

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

Example 3.273. Show quotas response: XML

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

This operation does not require a request body.

3.32.2.2. Response

Example 3.274. Update quota response: JSON

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

Example 3.275. Update quota response: XML

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

This operation does not return a response body.

3.32.3. Get default quotas

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

Normal response codes: 200

3.32.3.1. Request

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

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

This operation does not require a request body.

3.32.3.2. Response

Example 3.276. Get default quotas response: JSON

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

Example 3.277. Get default quotas response: XML

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

```
</quota_set>
```

This operation does not return a response body.

3.32.4. Show quotas for user

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

Normal response codes: 200

3.32.4.1. Request

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

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

This operation does not require a request body.

3.32.4.2. Response

Example 3.278. Show quotas for user response: JSON

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

Example 3.279. Show quotas for user response: XML

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

```
<key_pairs>100</key_pairs>  
<metadata_items>128</metadata_items>  
<ram>51200</ram>  
<security_group_rules>20</security_group_rules>  
<security_groups>10</security_groups>  
</quota_set>
```

This operation does not return a response body.

3.32.5. Update quotas for user

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

Normal response codes: 200

3.32.5.1. Request

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

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

Example 3.280. Update quotas for user request: JSON

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

Example 3.281. Update quotas for user request: XML

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

This operation does not require a request body.

3.32.5.2. Response

Example 3.282. Update quotas for user response: JSON

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

```
    "ram": 51200,  
    "security_group_rules": 20,  
    "security_groups": 10  
  }  
}
```

Example 3.283. Show quotas for user response: XML

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

This operation does not return a response body.

3.32.6. Show quota details for user

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

Normal response codes: 200

3.32.6.1. Request

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

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

This operation does not require a request body.

3.32.6.2. Response

Example 3.284. Show quota details for user response: JSON

```
{
  "quota_set": {
    "cores": {
      "in_use": 0,
      "limit": 20,
      "reserved": 0
    },
    "fixed_ips": {
      "in_use": 0,
      "limit": -1,
      "reserved": 0
    },
    "floating_ips": {
      "in_use": 0,
      "limit": 10,
      "reserved": 0
    },
    "injected_files": {
      "in_use": 0,
      "limit": 5,
      "reserved": 0
    },
    "instances": {
      "in_use": 0,
      "limit": 10,
      "reserved": 0
    },
    "key_pairs": {
      "in_use": 0,
```

```

        "limit":100,
        "reserved":0
    },
    "metadata_items":{
        "in_use":0,
        "limit":128,
        "reserved":0
    },
    "ram":{
        "in_use":0,
        "limit":51200,
        "reserved":0
    },
    "security_groups":{
        "in_use":0,
        "limit":10,
        "reserved":0
    },
    "injected_file_content_bytes":{
        "in_use":0,
        "limit":10240,
        "reserved":0
    },
    "injected_file_path_bytes":{
        "in_use":0,
        "limit":255,
        "reserved":0
    },
    "security_group_rules":{
        "in_use":0,
        "limit":20,
        "reserved":0
    }
}
}
}

```

3.33. Server rescue and unrescue (os-rescue)

Put a server into rescue mode or unrescue a server in rescue mode.

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Puts a server in rescue mode. Changes status to RESCUE.
POST	/v2/{tenant_id}/servers/{server_id}/action	Unrescues a server.

3.33.1. Rescue server

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/action	Puts a server in rescue mode. Changes status to RESCUE.

Normal response codes: 200

3.33.1.1. Request

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

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

Example 3.285. Rescue server: JSON request

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

Example 3.286. Rescue server: XML request

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

This operation does not require a request body.

3.33.1.2. Response

Example 3.287. Rescue server: JSON response

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

Example 3.288. Rescue server: XML response

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

This operation does not return a response body.

3.33.2. Unrescue server

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

Normal response codes: 202

3.33.2.1. Request

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

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

Example 3.289. Unrescue server: JSON request

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

Example 3.290. Unrescue server: XML request

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

This operation does not require a request body.

3.34. Rules for default security group (os-security-group-default-rules)

List, show information for, and create default security group rules.

Method	URI	Description
GET	/v2/{tenant_id}/os-security-group-rules	Lists default security group rules.
POST	/v2/{tenant_id}/os-security-group-rules	Creates a default security group rule.
GET	/v2/{tenant_id}/os-security-group-rules/{security_group_rule_id}	Shows information for a specified security group rule.

3.34.1. List default security group rules

Method	URI	Description
GET	/v2/{tenant_id}/os-security-group-rules	Lists default security group rules.

Normal response codes: 200

3.34.1.1. Request

This table shows the URI parameters for the list default security group rules request:

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

This operation does not require a request body.

3.34.1.2. Response

Example 3.291. 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
    }
  ]
}
```

Example 3.292. List default security group rules: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group_default_rules xmlns="http://docs.openstack.org/compute/api/v1.1">
  <security_group_default_rule id="1">
    <ip_protocol>TCP</ip_protocol>
    <from_port>80</from_port>
    <to_port>80</to_port>
    <ip_range>
      <cidr>10.10.10.0/24</cidr>
    </ip_range>
  </security_group_default_rule>
</security_group_default_rules>
```

This operation does not return a response body.

3.34.2. Create default security group rule

Method	URI	Description
POST	/v2/{tenant_id}/os-security-group-rules	Creates a default security group rule.

Normal response codes: 200

3.34.2.1. Request

This table shows the URI parameters for the create default security group rule request:

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

Example 3.293. Create default security group rule: JSON request

```
{
  "security_group_default_rule": {
    "ip_protocol": "TCP",
    "from_port": "80",
    "to_port": "80",
    "cidr": "10.10.12.0/24"
  }
}
```

Example 3.294. Create default security group rule: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group_default_rule>
  <ip_protocol>TCP</ip_protocol>
  <from_port>80</from_port>
  <to_port>80</to_port>
  <cidr>10.10.12.0/24</cidr>
</security_group_default_rule>
```

This operation does not require a request body.

3.34.2.2. Response

Example 3.295. 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
  }
}
```

Example 3.296. Create default security group rule: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group_default_rule xmlns="http://docs.openstack.org/compute/api/v1.1" id="1">
  <ip_protocol>TCP</ip_protocol>
  <from_port>80</from_port>
  <to_port>80</to_port>
  <ip_range>
    <cidr>10.10.10.0/24</cidr>
  </ip_range>
</security_group_default_rule>
```

This operation does not return a response body.

3.34.3. Show default security group rule information

Method	URI	Description
GET	/v2/{tenant_id}/os-security-group-rules/{security_group_rule_id}	Shows information for a specified security group rule.

Normal response codes: 200

3.34.3.1. Request

This table shows the URI parameters for the show default security group rule information request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{security_group_rule_id}	Uuid	The unique identifier of the security group rule.

This operation does not require a request body.

3.34.3.2. Response

Example 3.297. Show default security group rule: JSON response

```
{
  "security_group_default_rule": {
    "id": 1,
    "from_port": 80,
    "to_port": 80,
    "ip_protocol": "TCP",
    "ip_range": {
      "cidr": "10.10.10.0/24"
    }
  }
}
```

Example 3.298. Show default security group rule: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group_default_rule xmlns="http://docs.openstack.org/compute/api/v1.1" id="1">
  <from_port>80</from_port>
  <to_port>80</to_port>
  <ip_protocol>TCP</ip_protocol>
  <ip_range>
    <cidr>10.10.10.0/24</cidr>
  </ip_range>
</security_group_default_rule>
```

This operation does not return a response body.

3.35. Security groups (os-security-groups)

List, show information for, create, and delete security groups.

Method	URI	Description
GET	/v2/{tenant_id}/os-security-groups	Lists security groups.
POST	/v2/{tenant_id}/os-security-groups	Creates a security group.
GET	/v2/{tenant_id}/os-security-groups/servers/{server_id}/os-security-groups	Lists security groups for a specified server.
GET	/v2/{tenant_id}/os-security-groups/{security_group_id}	Shows information for a specified security group.
DELETE	/v2/{tenant_id}/os-security-groups/{security_group_id}	Deletes a specified security group.

3.35.1. List security groups

Method	URI	Description
GET	/v2/{tenant_id}/os-security-groups	Lists security groups.

Normal response codes: 200

3.35.1.1. Request

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

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

This operation does not require a request body.

3.35.1.2. Response

Example 3.299. List security groups: JSON response

```
{
  "security_groups": [
    {
      "description": "default",
      "id": 1,
      "name": "default",
      "rules": [],
      "tenant_id": "openstack"
    }
  ]
}
```

Example 3.300. List security group: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_groups xmlns="http://docs.openstack.org/compute/api/v1.1">
  <security_group tenant_id="openstack" id="1" name="default">
    <description>default</description>
    <rules/>
  </security_group>
</security_groups>
```

This operation does not return a response body.

3.35.2. Create security group

Method	URI	Description
POST	/v2/{tenant_id}/os-security-groups	Creates a security group.

Normal response codes: 200

3.35.2.1. Request

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

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

Example 3.301. Create security group: JSON request

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

Example 3.302. Create security group: XML request

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

This operation does not require a request body.

3.35.2.2. Response

Example 3.303. Create security group: JSON response

```
{
  "security_group": {
    "description": "description",
    "id": 2,
    "name": "test",
    "rules": [],
    "tenant_id": "openstack"
  }
}
```

Example 3.304. Create security group: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group xmlns="http://docs.openstack.org/compute/api/v1.1" tenant_id=
"openstack" id="2" name="test">
  <description>
    description
  </description>
</rules/>
```

```
</security_group>
```

This operation does not return a response body.

3.35.3. List security groups by server

Method	URI	Description
GET	/v2/{tenant_id}/os-security-groups/servers/{server_id}/os-security-groups	Lists security groups for a specified server.

Normal response codes: 200

3.35.3.1. Request

This table shows the URI parameters for the list security groups by server request:

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{server_id}	UUID	The UUID for the server of interest to you.

This operation does not require a request body.

3.35.3.2. Response

Example 3.305. List security groups by server: JSON response

```
{
  "security_groups": [
    {
      "description": "default",
      "id": 1,
      "name": "default",
      "rules": [],
      "tenant_id": "openstack"
    }
  ]
}
```

Example 3.306. List security groups by server: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_groups xmlns="http://docs.openstack.org/compute/api/v1.1">
  <security_group tenant_id="openstack" id="1" name="default">
    <description>default</description>
    <rules/>
  </security_group>
</security_groups>
```

This operation does not return a response body.

3.35.4. Show security group information

Method	URI	Description
GET	/v2/{tenant_id}/os-security-groups/{security_group_id}	Shows information for a specified security group.

Normal response codes: 200

3.35.4.1. Request

This table shows the URI parameters for the show security group information request:

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

This operation does not require a request body.

3.35.4.2. Response

Example 3.307. Show security group: JSON response

```
{
  "security_group": {
    "description": "default",
    "id": 1,
    "name": "default",
    "rules": [],
    "tenant_id": "openstack"
  }
}
```

Example 3.308. Show security group: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<security_group xmlns="http://docs.openstack.org/compute/api/v1.1" tenant_id=
"openstack" id="1" name="default">
  <description>default</description>
  <rules/>
</security_group>
```

This operation does not return a response body.

3.35.5. Delete security group

Method	URI	Description
DELETE	/v2/{tenant_id}/os-security-groups/{security_group_id}	Deletes a specified security group.

Normal response codes: 202

3.35.5.1. Request

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

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

This operation does not require a request body.

3.36. Server password (os-server-password)

Get and reset the encrypted admin password set through the metadata service.

Method	URI	Description
GET	/v2/servers/{server_id}/os-server-password	Gets the administrative password for a specified server.
DELETE	/v2/servers/{server_id}/os-server-password	Clears the encrypted copy of the password in the metadata server. This is done after the client has retrieved the password and knows it doesn't need it in the metadata server anymore. The password for the server remains the same.

3.36.1. Get server password

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

Normal response codes: 200

3.36.1.1. Request

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

Name	Type	Description
{server_id}	UUID	The UUID for the server of interest to you.

This operation does not require a request body.

3.36.1.2. Response

Example 3.309. Get server password: JSON response

```
{
  "password": "xlozO3wLCBRWAA2yDjCCVx8vwNPypxnypmRYDa/zErlQ+EzPe1S/
Gz6nfmC52mOlOSCRuUOmG7kqqge jPof6M7bOezS387z jQ4LSvvwp28zUknzy4YzfFGhnHAdai3TxUJ26pfQCYrq8UTzm
I1K2LsuiplxSJR7Wdke4zNXJjHHP2RfYsVbZ/k9ANu+Nz4iIH8/7Cacud/
pphH7EjrY6a4RZnrjQskrhKYed0YERpotyjYk1eDtRe72GrSiXteqCM4biaQ5w3ruS+AcX//
PXk3uJ5kC7d67fPXaVz4WaQRYMg=="
}
```

Example 3.310. Get server password: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<password>xlozO3wLCBRWAA2yDjCCVx8vwNPypxnypmRYDa/zErlQ+EzPe1S/
Gz6nfmC52mOlOSCRuUOmG7kqqge jPof6M7bOezS387z jQ4LSvvwp28zUknzy4YzfFGhnHAdai3TxUJ26pfQCYrq8UTzm
I1K2LsuiplxSJR7Wdke4zNXJjHHP2RfYsVbZ/k9ANu+Nz4iIH8/7Cacud/
pphH7EjrY6a4RZnrjQskrhKYed0YERpotyjYk1eDtRe72GrSiXteqCM4biaQ5w3ruS+AcX//
PXk3uJ5kC7d67fPXaVz4WaQRYMg==</password>
```

This operation does not return a response body.


```

QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
ZSBwYXR0ZXJucyBiZWZlbnMgYXN5b3VkeW5kIHRv
dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlmCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
b25zLiINCg0KLVJpY2hhcmQgQmFjaA==
</file>
</personality>
</server>

```

This operation does not require a request body.

3.36.2.2. Response

Example 3.313. Clear server password: JSON response

```

{
  "server": {
    "adminPass": "78AtBtuxTqZV",
    "id": "66fd64e1-de18-4506-bfb6-b5e73ef78a43",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/
66fd64e1-de18-4506-bfb6-b5e73ef78a43",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/
66fd64e1-de18-4506-bfb6-b5e73ef78a43",
        "rel": "bookmark"
      }
    ]
  }
}

```

Example 3.314. Clear server password: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1" id="b68e3354-0b1a-4e92-a664-8b332cff27f5" adminPass=
"sLV7uLzmg0Hu">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/
b68e3354-0b1a-4e92-a664-8b332cff27f5" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/
b68e3354-0b1a-4e92-a664-8b332cff27f5" rel="bookmark"/>
</server>

```

This operation does not return a response body.

3.37. Server shelve (os-server-shelve)

Shelve a running server.

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

3.37.1. Shelve server

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

Normal response codes: 202

3.37.1.1. Request

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

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

Example 3.315. Shelve server: JSON request

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

Example 3.316. Shelve server: XML request

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

This operation does not require a request body.

3.38. Server start and stop (os-server-start-stop)

Start a stopped server or stop a running server.

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

3.38.1. Start server

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

Normal response codes: 202

3.38.1.1. Request

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

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

Example 3.317. Start server: JSON request

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

Example 3.318. Start server: XML request

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

This operation does not require a request body.

3.38.2. Stop server

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

Normal response codes: 202

3.38.2.1. Request

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

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

Example 3.319. Stop server: JSON request

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

Example 3.320. Stop server: XML request

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

This operation does not require a request body.

3.39. Manage services (os-services)

List, enable, and disable Compute services in all hosts.

Method	URI	Description
GET	/v2/{tenant_id}/os-services	Lists running services.
PUT	/v2/{tenant_id}/os-services/enable	Enables scheduling for a service.
PUT	/v2/{tenant_id}/os-services/disable	Disables scheduling for a service.
PUT	/v2/{tenant_id}/os-services/disable-log-reason	Logs information to the service table about why a service was disabled.
GET	/v2/{tenant_id}/os-services/detail	Lists disabled services. If information exists, includes reasons why services were disabled.

3.39.1. List services

Method	URI	Description
GET	/v2/{tenant_id}/os-services	Lists running services.

Normal response codes: 200

3.39.1.1. Request

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

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

This operation does not require a request body.

3.39.1.2. Response

Example 3.321. List services: JSON response

```
{
  "services": [
    {
      "binary": "nova-scheduler",
      "host": "host1",
      "state": "up",
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:02.000000",
      "zone": "internal"
    },
    {
      "binary": "nova-compute",
      "host": "host1",
      "state": "up",
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:05.000000",
      "zone": "nova"
    },
    {
      "binary": "nova-scheduler",
      "host": "host2",
      "state": "down",
      "status": "enabled",
      "updated_at": "2012-09-19T06:55:34.000000",
      "zone": "internal"
    },
    {
      "binary": "nova-compute",
      "host": "host2",
      "state": "down",
      "status": "disabled",
      "updated_at": "2012-09-18T08:03:38.000000",
      "zone": "nova"
    }
  ]
}
```

```
}
```

3.39.2. Enable scheduling for a service

Method	URI	Description
PUT	/v2/{tenant_id}/os-services/enable	Enables scheduling for a service.

Normal response codes: 200

3.39.2.1. Request

This table shows the URI parameters for the enable scheduling for a service request:

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

Example 3.322. Enable scheduling for a service: JSON request

```
{
  "host": "host1",
  "binary": "nova-compute"
}
```

3.39.2.2. Response

Example 3.323. Enable scheduling for a service: JSON response

```
{
  "service": {
    "host": "host1",
    "binary": "nova-compute",
    "status": "enabled"
  }
}
```

3.39.3. Disable scheduling for a service

Method	URI	Description
PUT	/v2/{tenant_id}/os-services/disable	Disables scheduling for a service.

Normal response codes: 200

3.39.3.1. Request

This table shows the URI parameters for the disable scheduling for a service request:

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

Example 3.324. Disable scheduling for a service: JSON request

```
{
  "host": "host1",
  "binary": "nova-compute"
}
```

3.39.3.2. Response

Example 3.325. Disable scheduling for a service: JSON response

```
{
  "service": {
    "host": "host1",
    "binary": "nova-compute",
    "status": "disabled"
  }
}
```

3.39.4. Log disabled service information

Method	URI	Description
PUT	/v2/{tenant_id}/os-services/disable-log-reason	Logs information to the service table about why a service was disabled.

Normal response codes: 200

3.39.4.1. Request

This table shows the URI parameters for the log disabled service information request:

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

Example 3.326. Log disabled service information: JSON request

```
{
  "host": "host1",
  "binary": "nova-compute",
  "disabled_reason": "test2"
}
```

Example 3.327. Log disabled service information: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<service host="host1" binary="nova-compute" disabled_reason="test2"/>
```

This operation does not require a request body.

3.39.4.2. Response

Example 3.328. Log disabled service information: JSON response

```
{
  "service": {
    "binary": "nova-compute",
    "host": "host1",
    "disabled_reason": "test2",
    "status": "disabled"
  }
}
```

Example 3.329. Log disabled service information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<service host="host1" binary="nova-compute" status="disabled" disabled_reason="test2" />
```

This operation does not return a response body.

3.39.5. List disabled services

Method	URI	Description
GET	/v2/{tenant_id}/os-services/detail	Lists disabled services. If information exists, includes reasons why services were disabled.

Normal response codes: 200

3.39.5.1. Request

This table shows the URI parameters for the list disabled services request:

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

This operation does not require a request body.

3.39.5.2. Response

Example 3.330. List disabled services: JSON response

```
{
  "services": [
    {
      "binary": "nova-scheduler",
      "host": "host1",
      "state": "up",
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:02.000000",
      "zone": "internal",
      "disabled_reason": "test1"
    },
    {
      "binary": "nova-compute",
      "host": "host1",
      "state": "up",
      "status": "disabled",
      "updated_at": "2012-10-29T13:42:05.000000",
      "zone": "nova",
      "disabled_reason": "test2"
    },
    {
      "binary": "nova-scheduler",
      "host": "host2",
      "state": "down",
      "status": "enabled",
      "updated_at": "2012-09-19T06:55:34.000000",
      "zone": "internal",
      "disabled_reason": ""
    },
    {
      "binary": "nova-compute",
      "host": "host2",
      "state": "down",
      "status": "disabled",

```

```

        "updated_at": "2012-09-18T08:03:38.000000",
        "zone": "nova",
        "disabled_reason": "test4"
    }
]
}

```

Example 3.331. List disabled services: XML response

```

<services>
  <service status="disabled" binary="nova-scheduler" zone="internal" state="up" host="host1" updated_at="2012-10-29T13:42:02.000000" disabled_reason="test1"/>
  <service status="disabled" binary="nova-compute" zone="nova" state="up" host="host1" updated_at="2012-10-29T13:42:05.000000" disabled_reason="test2"/>
  <service status="enabled" binary="nova-scheduler" zone="internal" state="down" host="host2" updated_at="2012-09-19T06:55:34.000000" disabled_reason=""/>
  <service status="disabled" binary="nova-compute" zone="nova" state="down" host="host2" updated_at="2012-09-18T08:03:38.000000" disabled_reason="test4"/>
</services>

```

This operation does not return a response body.

3.40. Usage reports (os-simple-tenant-usage)

Report usage statistics on compute and storage resources.

Method	URI	Description
GET	/v2/{tenant_id}/os-simple-tenant-usage	Lists usage information for all tenants.
GET	/v2/{tenant_id}/os-simple-tenant-usage/{tenant_id}	Gets usage information for a tenant.

3.40.1. List usage information for all tenants

Method	URI	Description
GET	/v2/{tenant_id}/os-simple-tenant-usage	Lists usage information for all tenants.

Normal response codes: 200

3.40.1.1. Request

This table shows the URI parameters for the list usage information for all tenants request:

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

3.40.1.2. Response

Example 3.332. List usage information for all tenants: JSON response

```
{
  "tenant_usages": [
    {
      "start": "2012-10-08T21:10:44.587336",
      "stop": "2012-10-08T22:10:44.587336",
      "tenant_id": "openstack",
      "total_hours": 1.0,
      "total_local_gb_usage": 1.0,
      "total_memory_mb_usage": 512.0,
      "total_vcpus_usage": 1.0
    }
  ]
}
```

Example 3.333. List usage information for all tenants: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<tenant_usages>
  <tenant_usage>
    <tenant_id>openstack</tenant_id>
    <total_local_gb_usage>1.0</total_local_gb_usage>
    <total_vcpus_usage>1.0</total_vcpus_usage>
    <total_memory_mb_usage>512.0</total_memory_mb_usage>
    <total_hours>1.0</total_hours>
    <start>2012-10-08 21:10:51.902640</start>
    <stop>2012-10-08 22:10:51.902640</stop>
    <server_usages/>
  </tenant_usage>
</tenant_usages>
```

This operation does not return a response body.

3.40.2. Get tenant usage information

Method	URI	Description
GET	/v2/{tenant_id}/os-simple-tenant-usage/{tenant_id}	Gets usage information for a tenant.

Normal response codes: 200

3.40.2.1. Request

This table shows the URI parameters for the get tenant usage information request:

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

3.40.2.2. Response

Example 3.334. Get tenant usage information: JSON response

```
{
  "tenant_usage": {
    "server_usages": [
      {
        "ended_at": null,
        "flavor": "ml.tiny",
        "hours": 1.0,
        "instance_id": "1f1deceb-17b5-4c04-84c7-e0d4499c8fe0",
        "local_gb": 1,
        "memory_mb": 512,
        "name": "new-server-test",
        "started_at": "2012-10-08T20:10:44.541277",
        "state": "active",
        "tenant_id": "openstack",
        "uptime": 3600,
        "vcpus": 1
      }
    ],
    "start": "2012-10-08T20:10:44.587336",
    "stop": "2012-10-08T21:10:44.587336",
    "tenant_id": "openstack",
    "total_hours": 1.0,
    "total_local_gb_usage": 1.0,
    "total_memory_mb_usage": 512.0,
    "total_vcpus_usage": 1.0
  }
}
```

Example 3.335. Get tenant usage information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<tenant_usage>
  <tenant_id>openstack</tenant_id>
  <total_local_gb_usage>1.0</total_local_gb_usage>
  <total_vcpus_usage>1.0</total_vcpus_usage>
</tenant_usage>
```

```
<total_memory_mb_usage>512.0</total_memory_mb_usage>
<total_hours>1.0</total_hours>
<start>2012-10-08 20:10:51.902640</start>
<stop>2012-10-08 21:10:51.902640</stop>
<server_usages>
  <server_usage>
    <instance_id>e4521f3b-d9ad-4454-be8a-e2732f0630ef</instance_id>
    <name>new-server-test</name>
    <hours>1.0</hours>
    <memory_mb>512</memory_mb>
    <local_gb>1</local_gb>
    <vcpus>1</vcpus>
    <tenant_id>openstack</tenant_id>
    <flavor>m1.tiny</flavor>
    <started_at>2012-10-08 20:10:51.854331</started_at>
    <ended_at>None</ended_at>
    <state>active</state>
    <uptime>3600</uptime>
  </server_usage>
</server_usages>
</tenant_usage>
```

This operation does not return a response body.

3.41. Virtual interfaces (os-virtual-interfaces)

List the virtual interfaces for a specified server instance.

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Lists the virtual interfaces for a specified instance.
GET	/v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Shows the virtual interface for for a specified instance. Includes the OS-EXT-VIF-NET:net_id attribute that shows to which network the interface is attached.

3.41.1. List virtual interfaces

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

Normal response codes: 202

3.41.1.1. Request

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

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

This operation does not require a request body.

3.41.1.2. Response

Example 3.336. List virtual interfaces: JSON response

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

Example 3.337. List virtual interfaces: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<virtual_interfaces xmlns="http://docs.openstack.org/compute/api/v1.1">
  <virtual_interface id="94edf7aa-565a-469a-8f45-656b4acf8229" mac_address=
"fa:16:3e:7d:31:9a"/>
</virtual_interfaces>
```

This operation does not return a response body.

3.41.2. Show virtual interface and attached network

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-virtual-interfaces	Shows the virtual interface for a specified instance. Includes the OS-EXT-VIF-NET:net_id attribute that shows to which network the interface is attached.

Normal response codes: 202

3.41.2.1. Request

This table shows the URI parameters for the show virtual interface and attached network request:

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

This operation does not require a request body.

3.41.2.2. Response

Example 3.338. Show virtual interface and attached network: JSON response

```
{
  "virtual_interfaces": [
    {
      "id": "cec8b9bb-5d22-4104-b3c8-4c35db3210a6",
      "mac_address": "fa:16:3e:3c:ce:6f",
      "OS-EXT-VIF-NET:net_id": "cec8b9bb-5d22-4104-b3c8-4c35db3210a7"
    }
  ]
}
```

Example 3.339. Show virtual interface and attached network: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<virtual_interfaces
  xmlns:OS-EXT-VIF-NET="http://docs.openstack.org/compute/ext/extended-
virtual-interfaces-net/api/v1.1"
  xmlns="http://docs.openstack.org/compute/api/v1.1">
  <virtual_interface id="94edf7aa-565a-469a-8f45-656b4acf8229"
    mac_address="fa:16:3e:7d:31:9a"
    OS-EXT-VIF-NET:net_id="94edf7aa-565a-469a-8f45-656b4acf8230"/>
</virtual_interfaces>
```

This operation does not return a response body.

3.42. Volume extension (os-volumes, os-snapshots)

Manage volumes and snapshots for use with the Compute API.

Method	URI	Description
GET	/v1.1/{tenant_id}/os-volumes	Lists the volumes associated with the account.
GET	/v1.1/{tenant_id}/os-volumes/ detail	Lists details for a specified volume.
POST	/v1.1/{tenant_id}/os-volumes/ {volume_id}	Creates a volume.
GET	/v1.1/{tenant_id}/os-volumes/ {volume_id}	Shows information for a specified volume.
DELETE	/v1.1/{tenant_id}/os-volumes/ {volume_id}	Deletes a specified volume.
GET	/v1.1/{tenant_id}/os-volume-types	Lists volume types.
GET	/v1.1/{tenant_id}/os-volume-types/ {volume_type_id}	Shows information for a specified volume type.
POST	/v1.1/{tenant_id}/os-snapshots	Creates a snapshot.
GET	/v1.1/{tenant_id}/os-snapshots	Lists snapshots.
GET	/v1.1/{tenant_id}/os-snapshots/ detail	Lists details for a specified snapshot.
GET	/v1.1/{tenant_id}/os-snapshots/ {snapshot_id}	Shows information for a specified snapshot.
DELETE	/v1.1/{tenant_id}/os-snapshots/ {snapshot_id}	Deletes a specified snapshot from the account.

3.42.1. List volumes

Method	URI	Description
GET	/v1.1/{tenant_id}/os-volumes	Lists the volumes associated with the account.

Normal response codes: 200

3.42.1.1. Request

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

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

This operation does not require a request body.

3.42.1.2. Response

Example 3.340. List volumes: JSON response

```
{
  "volumes": [
    {
      "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
      "display_name": "vol-001",
      "display_description": "Another volume.",
      "size": 30,
      "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
      "metadata": {"contents": "junk"},
      "availability_zone": "us-east1",
      "snapshot_id": null,
      "attachments": [],
      "created_at": "2012-02-14T20:53:07Z"
    },
    {
      "id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
      "display_name": "vol-002",
      "display_description": "Yet another volume.",
      "size": 25,
      "volume_type": "96c3bda7-c82a-4f50-be73-ca7621794835",
      "metadata": {},
      "availability_zone": "us-east2",
      "snapshot_id": null,
      "attachments": [],
      "created_at": "2012-03-15T19:10:03Z"
    }
  ]
}
```

Example 3.341. List volumes: XML response

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

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

This operation does not return a response body.

3.42.2. List details for volumes

Method	URI	Description
GET	/v1.1/{tenant_id}/os-volumes/detail	Lists details for a specified volume.

Normal response codes: 200

3.42.2.1. Request

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

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

This operation does not require a request body.

3.42.2.2. Response

Example 3.342. List details for volumes: JSON response

```
{
  "volumes": [
    {
      "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
      "display_name": "vol-001",
      "display_description": "Another volume.",
      "size": 30,
      "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
      "metadata": {"contents": "junk"},
      "availability_zone": "us-east1",
      "snapshot_id": null,
      "attachments": [],
      "created_at": "2012-02-14T20:53:07Z"
    },
    {
      "id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
      "display_name": "vol-002",
      "display_description": "Yet another volume.",
      "size": 25,
      "volume_type": "96c3bda7-c82a-4f50-be73-ca7621794835",
      "metadata": {},
      "availability_zone": "us-east2",
      "snapshot_id": null,
      "attachments": [],
      "created_at": "2012-03-15T19:10:03Z"
    }
  ]
}
```

Example 3.343. List details for volumes: XML response

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

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

This operation does not return a response body.

3.42.3. Create volume

Method	URI	Description
POST	/v1.1/{tenant_id}/os-volumes/{volume_id}	Creates a volume.

Normal response codes: 201

3.42.3.1. Request

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

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

Example 3.344. 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"
  }
}
```

Example 3.345. Create volume: XML request

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

This operation does not require a request body.

3.42.3.2. Response

Example 3.346. Create volume: JSON response

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

```
    "size": 30,  
    "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",  
    "metadata": {"contents": "junk"},  
    "availability_zone": "us-east1",  
    "snapshot_id": null,  
    "attachments": [],  
    "created_at": "2012-02-14T20:53:07Z"  
  }  
}
```

Example 3.347. Create volume: XML response

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

This operation does not return a response body.

3.42.4. Show volume information

Method	URI	Description
GET	/v1.1/{tenant_id}/os-volumes/{volume_id}	Shows information for a specified volume.

Normal response codes: 200

3.42.4.1. Request

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

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

This operation does not require a request body.

3.42.4.2. Response

Example 3.348. Show volume information: JSON response

```
{
  "volume": {
    "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "display_name": "vol-001",
    "display_description": "Another volume.",
    "size": 30,
    "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
    "metadata": {"contents": "junk"},
    "availability_zone": "us-east1",
    "snapshot_id": null,
    "attachments": [],
    "created_at": "2012-02-14T20:53:07Z"
  }
}
```

Example 3.349. Show volume information: XML response

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

This operation does not return a response body.

3.42.5. Delete volume

Method	URI	Description
DELETE	/v1.1/{tenant_id}/os-volumes/{volume_id}	Deletes a specified volume.

Normal response codes: 202

3.42.5.1. Request

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

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

This operation does not require a request body.

3.42.5.2. Response

Example 3.350. Delete volume: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Date: Fri, 02 Dec 2011 00:39:32 GMT
```

This operation does not return a response body.

3.42.6. List volume types

Method	URI	Description
GET	/v1.1/{tenant_id}/os-volume-types	Lists volume types.

Normal response codes: 200

3.42.6.1. Request

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

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

This operation does not require a request body.

3.42.6.2. Response

Example 3.351. List volume types: JSON response

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

Example 3.352. List volume types: XML response

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

This operation does not return a response body.

3.42.7. Show volume type

Method	URI	Description
GET	/v1.1/{tenant_id}/os-volume-types/{volume_type_id}	Shows information for a specified volume type.

Normal response codes: 200

3.42.7.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{volume_type_id}	String	The unique identifier for a volume type.

This operation does not require a request body.

3.42.7.2. Response

Example 3.353. Show volume type: JSON response

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

Example 3.354. Show volume type: XML response

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

This operation does not return a response body.

3.42.8. Create snapshot

Method	URI	Description
POST	/v1.1/{tenant_id}/os-snapshots	Creates a snapshot.

Normal response codes: 201

3.42.8.1. Request

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

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

Example 3.355. Create snapshot: JSON request

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

Example 3.356. Create snapshot: XML request

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

This operation does not require a request body.

3.42.8.2. Response

Example 3.357. Create snapshot: JSON response

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

Example 3.358. Create snapshot: XML response

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

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

This operation does not return a response body.

3.42.9. List snapshots

Method	URI	Description
GET	/v1.1/{tenant_id}/os-snapshots	Lists snapshots.

Normal response codes: 200

3.42.9.1. Request

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

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

This operation does not require a request body.

3.42.9.2. Response

Example 3.359. List snapshots: JSON response

```
{
  "snapshots": [
    {
      "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
      "display_name": "snap-001",
      "display_description": "Daily backup",
      "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
      "status": "available",
      "size": 30,
      "created_at": "2012-02-29T03:50:07Z"
    },
    {
      "id": "e479997c-650b-40a4-9dfe-77655818b0d2",
      "display_name": "snap-002",
      "display_description": "Weekly backup",
      "volume_id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
      "status": "available",
      "size": 25,
      "created_at": "2012-03-19T01:52:47Z"
    }
  ]
}
```

Example 3.360. List snapshots: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshots xmlns="http://docs.openstack.org/volume/api/v1">
  <snapshot id="3fbbcccf-d058-4502-8844-6feeffdf4cb5"
    display_name="snap-001"
    display_description="Daily backup"
    volume_id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
    status="available"
    size="30"
    created_at="2012-02-29T03:50:07Z" />
  <snapshot id="e479997c-650b-40a4-9dfe-77655818b0d2"
```

```
display_name="snap-002"  
display_description="Weekly backup"  
volume_id="76b8950a-8594-4e5b-8dce-0dfa9c696358"  
status="available"  
size="25"  
created_at="2012-03-19T01:52:47Z" />  
</snapshots>
```

This operation does not return a response body.

3.42.10. List details for snapshots

Method	URI	Description
GET	/v1.1/{tenant_id}/os-snapshots/detail	Lists details for a specified snapshot.

Normal response codes: 200

3.42.10.1. Request

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

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

This operation does not require a request body.

3.42.10.2. Response

Example 3.361. List details for snapshots: JSON response

```
{
  "snapshots": [
    {
      "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
      "display_name": "snap-001",
      "display_description": "Daily backup",
      "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
      "status": "available",
      "size": 30,
      "created_at": "2012-02-29T03:50:07Z"
    },
    {
      "id": "e479997c-650b-40a4-9dfe-77655818b0d2",
      "display_name": "snap-002",
      "display_description": "Weekly backup",
      "volume_id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
      "status": "available",
      "size": 25,
      "created_at": "2012-03-19T01:52:47Z"
    }
  ]
}
```

Example 3.362. List details for snapshots: XML response

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

```
<snapshot id="e479997c-650b-40a4-9dfe-77655818b0d2"  
  display_name="snap-002"  
  display_description="Weekly backup"  
  volume_id="76b8950a-8594-4e5b-8dce-0dfa9c696358"  
  status="available"  
  size="25"  
  created_at="2012-03-19T01:52:47Z" />  
</snapshots>
```

This operation does not return a response body.

3.42.11. Show snapshot

Method	URI	Description
GET	/v1.1/{tenant_id}/os-snapshots/{snapshot_id}	Shows information for a specified snapshot.

Normal response codes: 200

3.42.11.1. Request

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

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

This operation does not require a request body.

3.42.11.2. Response

Example 3.363. Show snapshot: JSON response

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

Example 3.364. Show snapshot: XML response

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

This operation does not return a response body.

3.42.12. Delete snapshot

Method	URI	Description
DELETE	/v1.1/{tenant_id}/os-snapshots/{snapshot_id}	Deletes a specified snapshot from the account.

This operation is asynchronous. You must list snapshots repeatedly to determine whether the snapshot was deleted.

Normal response codes: 202

3.42.12.1. Request

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

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

This operation does not require a request body.

3.42.12.2. Response

Example 3.365. Delete snapshot: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Date: Mon, 05 Dec 2011 16:23:10 GMT
```

This operation does not return a response body.

3.43. Volume attachments (os-volume_attachments)

Attach volumes created through the volume API to server instances. Also, list volume attachments for a server instance, get volume details for a volume attachment, and delete a volume attachment.

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Attaches a volume to the specified server.
GET	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Lists the volume attachments for a specified server.
GET	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Shows details for the specified volume attachment.
DELETE	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Deletes the specified volume attachment from a specified server.

3.43.1. Attach volume

Method	URI	Description
POST	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Attaches a volume to the specified server.

Normal response codes: 202

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

3.43.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.
{volumeId}	String	ID of the volume to attach.
{device}	String	Name of the device such as, /dev/vdb. Use "auto" for auto-assign (if supported).
{volumeAttachment}	String	A dictionary representation of a volume attachment.

Example 3.366. Attach volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volumeAttachment volumeId="a26887c6-c47b-4654-abb5-dfadf7d3f803" device="/dev/vdd" />
```

Example 3.367. Attach volume: JSON request

```
{
  "volumeAttachment": {
    "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
    "device": "/dev/vdd"
  }
}
```

3.43.1.2. Response

Example 3.368. Attach volume: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumeAttachment device="/dev/vdd" serverId="20f0cb44-7b00-4019-a612-364777cd2931" id="a26887c6-c47b-4654-abb5-dfadf7d3f803" volumeId="a26887c6-c47b-4654-abb5-dfadf7d3f803"/>
```

Example 3.369. Attach volume: JSON response

```
{
  "volumeAttachment": {
    "device": "/dev/vdd",
```

```
    "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",  
    "serverId": "0c92f3f6-c253-4c9b-bd43-e880a8d2eb0a",  
    "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"  
  }  
}
```

3.43.2. List volume attachments

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments	Lists the volume attachments for a specified server.

Normal response codes: 202

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

3.43.2.1. Request

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

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

This operation does not require a request body.

3.43.2.2. Response

Example 3.370. List volume attachments: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumeAttachments>
  <volumeAttachment device="/dev/sdd" serverId=
"4335bab6-6086-4247-8274-8b8b048edaaa" id="a26887c6-c47b-4654-abb5-
dfadf7d3f803" volumeId="a26887c6-c47b-4654-abb5-dfadf7d3f803" />
  <volumeAttachment device="/dev/sdc" serverId=
"4335bab6-6086-4247-8274-8b8b048edaaa" id="a26887c6-c47b-4654-abb5-
dfadf7d3f804" volumeId="a26887c6-c47b-4654-abb5-dfadf7d3f804" />
</volumeAttachments>
```

Example 3.371. List volume attachments: JSON response

```
{
  "volumeAttachments": [
    {
      "device": "/dev/sdd",
      "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
      "serverId": "4d8c3732-a248-40ed-bebc-539a6ffd25c0",
      "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
    },
    {
      "device": "/dev/sdc",
      "id": "a26887c6-c47b-4654-abb5-dfadf7d3f804",
      "serverId": "4d8c3732-a248-40ed-bebc-539a6ffd25c0",
      "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f804"
    }
  ]
}
```


3.43.3. Show volume attachment details

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Shows details for the specified volume attachment.

Normal response codes: 202

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

3.43.3.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.
{attachment_id}	String	Volume attachment ID.

This operation does not require a request body.

3.43.3.2. Response

Example 3.372. Show volume attachment details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumeAttachment device="/dev/sdd" serverId="20f12907-3993-44f7-a680-
d51e2ceedbd9" id="a26887c6-c47b-4654-abb5-dfadf7d3f803" volumeId="a26887c6-
c47b-4654-abb5-dfadf7d3f803"/>
```

Example 3.373. Show volume attachment details: JSON response

```
{
  "volumeAttachment": {
    "device": "/dev/sdd",
    "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
    "serverId": "2390fb4d-1693-45d7-b309-e29c4af16538",
    "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
  }
}
```

3.43.4. Delete volume attachment

Method	URI	Description
DELETE	/v2/{tenant_id}/servers/{server_id}/os-volume_attachments/{attachment_id}	Deletes the specified volume attachment from a specified server.

Normal response codes: 202

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

3.43.4.1. Request

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

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{server_id}	UUID	The UUID for the server of interest to you.
{attachment_id}	String	Volume attachment ID.

This operation does not require a request body.

3.44. Servers with block device mapping format (servers)

Create a server with a block device mapping.

Method	URI	Description
GET	/v2/{tenant_id}/servers{?changesince,image,flavor,name,marker,limit,status,host}	Lists IDs, names, and links for all servers.
POST	/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}	Creates a server with a block device mapping.

3.44.1. List servers

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

Normal response codes: 200, 203

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

3.44.1.1. Request

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

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

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

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

3.44.1.2. Response

Example 3.374. List servers: JSON response

```
{
  "servers": [
    {
```

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

Example 3.375. List servers: XML response

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

3.44.2. Create server

Method	URI	Description
POST	/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}	Creates a server with a block device mapping.

Normal response codes: 202

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

3.44.2.1. Request

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

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

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

Name	Type	Description
security_group	String <i>(Required)</i>	The name of the security group. If blank, the server is created in the default security group.
user_data	String <i>(Optional)</i>	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String <i>(Optional)</i>	The availability zone in which to launch the server.

Example 3.376. Create server: JSON request

```
{
  "server" : {
    "name" : "new-server-test",
    "imageRef" : "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
    "flavorRef" : "http://openstack.example.com/openstack/flavors/1",
    "metadata" : {
      "My Server Name" : "Apache1"
    },
    "personality" : [
      {
        "path" : "/etc/banner.txt",
        "contents" :
" ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb2"
        }
    ],
    "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"
    }
]
}
}

```

Example 3.377. Create server: XML request

```

<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1" imageRef="http://
openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b"
  flavorRef="http://openstack.example.com/openstack/flavors/1" name="new-
server-test">
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAGICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
      dCBtb3Z1cyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
      IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVz
      c2lubi4uLnRoXMGaXMgdGhlIHBSYWNlIHRvIGdvIG5vdy4g
      QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
      ZSBwYXR0ZXJucyBiZWwhpbmQgYWxsIGNsb3VkcycwYmV5kIHlv
      dSB3aWxsIGtub3csIHRvbywgZ2hlbiB5b3UgbGlmZCB5b3Vy
      c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
      b25zLiINCgOKLVJpY2hhcmQgQmFjaA==
    </file>
  </personality>
  <block_device_mapping_v2>
    <mapping device_name="/dev/sdb1" source_type="blank" destination_type=
"local" delete_on_termination="True" guest_format="swap" boot_index="-1"></
mapping>
    <mapping device_name="/dev/sda1" source_type="volume" destination_type=
"volume" uuid="fake-volume-id-1" boot_index="0"></mapping>
  </block_device_mapping_v2>
</server>

```

3.44.2.2. Response

Example 3.378. Create server: JSON response

```

{
  "server": {
    "adminPass": "N4x7wFX6iN8D",
    "id": "babdlaf0-4fc6-4529-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"
        }
      ]
    }
  }
}

```

Example 3.379. Create server: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1" id="fdbce07b-097e-4ab1-8141-b1c847861aa1" adminPass=
"zA62GVkFvN74">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/
fdbce07b-097e-4ab1-8141-b1c847861aa1" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/
fdbce07b-097e-4ab1-8141-b1c847861aa1" rel="bookmark"/>
</server>

```

3.45. Server OS-EXT-IPS-MAC:mac_addr extended attribute (servers)

Add OS-EXT-IPS-MAC:mac_addr extended attribute when you create, show information for, or list servers.

Method	URI	Description
POST	/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}	Creates a server with the OS-EXT-IPS-MAC:mac_addr extended attribute.
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.
GET	/v2/{tenant_id}/servers/detail	Lists details for all servers. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.

3.45.1. Create server with OS-EXT-IPS-MAC:mac_addr extended attribute

Method	URI	Description
POST	/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}	Creates a server with the OS-EXT-IPS-MAC:mac_addr extended attribute.

Normal response codes: 202

3.45.1.1. Request

This table shows the URI parameters for the create server with os-ext-ips-mac:mac_addr extended attribute request:

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

This table shows the query parameters for the create server with os-ext-ips-mac:mac_addr extended attribute request:

Name	Type	Description
security_group	String <i>(Required)</i>	The name of the security group. If blank, the server is created in the default security group.
user_data	String <i>(Optional)</i>	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String <i>(Optional)</i>	The availability zone in which to launch the server.

Example 3.380. Create server with OS-EXT-IPS-MAC:mac_addr extended attribute: JSON request

```
{
  "server" : {
    "name" : "new-server-test",
    "imageRef" : "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
    "flavorRef" : "http://openstack.example.com/openstack/flavors/1",
    "metadata" : {
      "My Server Name" : "Apache1"
    },
    "personality" : [
      {
        "path" : "/etc/banner.txt",
        "contents" :
          "ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcYBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb2"
      }
    ]
  }
}
```

```
}

```

Example 3.381. Create server with OS-EXT-IPS-MAC:mac_addr extended attribute: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1" imageRef="http://
openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b"
flavorRef="http://openstack.example.com/openstack/flavors/1" name="new-
server-test">
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
      dCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
      IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGlhcHVz
      c2lvbi4uLnRoXMGaXMgdGhlIHBSYWNlIHRvIGdvIG5vdy4g
      QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
      ZSBwYXR0ZXJucyBiZWVhbmQgYWxsIGNsb3VkcycwYmV5kIHlv
      dSB3aWxsIGtub3csIHRvbywgZ2hlbiB5b3UgbGlmZCB5b3Vy
      c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
      b25zLiINCg0KLVJpY2hhcmQgQmFjaA==
    </file>
  </personality>
</server>
```

This operation does not require a request body.

3.45.1.2. Response

Example 3.382. Create server with OS-EXT-IPS-MAC:mac_addr extended attribute: JSON response

```
{
  "server": {
    "adminPass": "zD7wDKTXiHsp",
    "id": "b44e5008-42f7-4048-b4c8-f40a29da88ba",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/
b44e5008-42f7-4048-b4c8-f40a29da88ba",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/
b44e5008-42f7-4048-b4c8-f40a29da88ba",
        "rel": "bookmark"
      }
    ]
  }
}
```

Example 3.383. Create server with OS-EXT-IPS-MAC:mac_addr extended attribute: XML response

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

```
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" id="752dd57d-933b-4a57-a0ae-4c3431c5abc7" adminPass="B2gvFFjBQCVQ">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/752dd57d-933b-4a57-a0ae-4c3431c5abc7" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/752dd57d-933b-4a57-a0ae-4c3431c5abc7" rel="bookmark"/>
</server>
```

This operation does not return a response body.

3.45.2. Show server information

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.

Normal response codes: 202

3.45.2.1. Request

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

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

This operation does not require a request body.

3.45.2.2. Response

Example 3.384. Show server information: JSON response

```
{
  "server": {
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4,
          "OS-EXT-IPS-MAC:mac_addr": "00:0c:29:e1:42:90"
        }
      ]
    },
    "created": "2013-02-07T18:46:28Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "4e2003eddbfdb1280c2618d04090bccd6773203b8da8347af0b2723d",
    "id": "dc7281f9-ee47-40b9-9950-9f73e7961caa",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    }
  }
}
```

```

    }
  ],
  "links": [
    {
      "href": "http://openstack.example.com/v2/openstack/servers/
dc7281f9-ee47-40b9-9950-9f73e7961caa",
      "rel": "self"
    },
    {
      "href": "http://openstack.example.com/openstack/servers/
dc7281f9-ee47-40b9-9950-9f73e7961caa",
      "rel": "bookmark"
    }
  ],
  "metadata": {
    "My Server Name": "Apache1"
  },
  "name": "new-server-test",
  "progress": 0,
  "status": "ACTIVE",
  "tenant_id": "openstack",
  "updated": "2013-02-07T18:46:29Z",
  "user_id": "fake"
}
}

```

Example 3.385. Show server information: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-EXT-IPS-MAC="http://docs.openstack.org/compute/
ext/extended_ips_mac/api/v1.1" xmlns:atom="http://www.w3.
org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/
v1.1" status="ACTIVE" updated="2013-02-07T18:46:29Z" hostId=
"068cc5e2de14b6e533a239c6eac0a0bdedcd57cab25450a6d3da43af" name="new-
server-test" created="2013-02-07T18:46:28Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="22e7cf4d-
ab7a-4a3d-9599-7d0dbaf9ed55">
  <image id="70a599e0-31e7-49b7-b260-868f441e862b">
    <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  </image>
  <flavor id="1">
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="private">
      <ip version="4" addr="192.168.0.3" OS-EXT-IPS-MAC:mac_addr=
"00:0c:29:e1:42:90"/>
    </network>
  </addresses>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/22e7cf4d-
ab7a-4a3d-9599-7d0dbaf9ed55" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/22e7cf4d-
ab7a-4a3d-9599-7d0dbaf9ed55" rel="bookmark"/>
</server>

```

This operation does not return a response body.

3.45.3. Get server details

Method	URI	Description
GET	/v2/{tenant_id}/servers/detail	Lists details for all servers. Includes the OS-EXT-IPS-MAC:mac_addr extended attribute.

Normal response codes: 202

3.45.3.1. Request

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

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

This operation does not require a request body.

3.45.3.2. Response

Example 3.386. Get server details: JSON response

```
{
  "servers": [
    {
      "accessIPv4": "",
      "accessIPv6": "",
      "addresses": {
        "private": [
          {
            "addr": "192.168.0.3",
            "version": 4,
            "OS-EXT-IPS-MAC:mac_addr": "00:0c:29:e1:42:90"
          }
        ]
      },
      "created": "2013-02-07T18:40:59Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "hostId":
      "fe866a4962fe3bdb6c2db9c8f7dcdb9555aca73387e72b5cb9c45bd3",
      "id": "76908712-653a-4d16-807e-d89d41435d24",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
            "rel": "bookmark"
          }
        ]
      }
    }
  ]
}
```

```

    }
  ],
  "links": [
    {
      "href": "http://openstack.example.com/v2/openstack/
servers/76908712-653a-4d16-807e-d89d41435d24",
      "rel": "self"
    },
    {
      "href": "http://openstack.example.com/openstack/servers/
76908712-653a-4d16-807e-d89d41435d24",
      "rel": "bookmark"
    }
  ],
  "metadata": {
    "My Server Name": "Apache1"
  },
  "name": "new-server-test",
  "progress": 0,
  "status": "ACTIVE",
  "tenant_id": "openstack",
  "updated": "2013-02-07T18:40:59Z",
  "user_id": "fake"
}
]
}

```

Example 3.387. Get server details: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:OS-EXT-IPS-MAC="http://docs.openstack.org/compute/ext/
extended_ips_mac/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1">
  <server status="ACTIVE" updated="2013-02-07T18:40:59Z" hostId=
"51a80e6ee89b638b2cb57eb4e39d89a725e07c8a698f4d8e256f8665" name=
"new-server-test" created="2013-02-07T18:40:59Z" userId="fake"
tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id=
"0337de6b-1d43-46c8-8804-35669f1dea9a">
  <image id="70a599e0-31e7-49b7-b260-868f441e862b">
    <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  </image>
  <flavor id="1">
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="private">
      <ip version="4" addr="192.168.0.3" OS-EXT-IPS-MAC:mac_addr=
"00:0c:29:e1:42:90"/>
    </network>
  </addresses>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/
0337de6b-1d43-46c8-8804-35669f1dea9a" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/
0337de6b-1d43-46c8-8804-35669f1dea9a" rel="bookmark"/>

```

```
</server>
</servers>
```

This operation does not return a response body.

3.46. Configuration drive (servers)

Extend servers and images with a configuration drive.

Method	URI	Description
POST	/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}	Creates a server with the configuration drive extended attribute.
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server including the configuration drive extended attribute.
GET	/v2/{tenant_id}/servers/{server_id}/detail	Lists details for all servers including the configuration drive extended attribute.


```
<server xmlns="http://docs.openstack.org/compute/api/v1.1" imageRef="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" flavorRef="http://openstack.example.com/openstack/flavors/1" name="new-server-test">
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
      dCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
      IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVz
      c2lvsbi4uLnRoXMGaXMgdGhlIHBSYWNlIHRvIGdvIG5vdy4g
      QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
      ZSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlmdCB5b3Vy
      c2V5ZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
      b25zLiINCgOKLVJpY2hhcmQgQmFjaA==
    </file>
  </personality>
</server>
```

This operation does not require a request body.

3.46.1.2. Response

Example 3.390. Create server with configuration drive: JSON response

```
{
  "server": {
    "adminPass": "am5LKVsBVQ4s",
    "id": "58da039c-dc81-4d8f-8688-a2f819e2f750",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/58da039c-dc81-4d8f-8688-a2f819e2f750",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/58da039c-dc81-4d8f-8688-a2f819e2f750",
        "rel": "bookmark"
      }
    ]
  }
}
```

Example 3.391. Create server with configuration drive: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" id="00bba779-f8ae-403e-901d-laf18bcb9187" adminPass="XJqvFkH62TZh">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/00bba779-f8ae-403e-901d-laf18bcb9187" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/00bba779-f8ae-403e-901d-laf18bcb9187" rel="bookmark"/>
</server>
```


This operation does not return a response body.

3.46.2. Get server information with configuration drive

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server including the configuration drive extended attribute.

Normal response codes: 202

3.46.2.1. Request

This table shows the URI parameters for the get server information with configuration drive request:

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

This operation does not require a request body.

3.46.2.2. Response

Example 3.392. Get server information with configuration drive: JSON response

```
{
  "server": {
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "config_drive": "",
    "created": "2013-02-04T13:17:50Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "8725fb615b191d8249a40f3e90d1efde88d914412e4edb2719176afd",
    "id": "dd3b0715-a3fc-43d8-bbd2-2720beb226fb",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [

```

```

        "href": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
        "rel": "bookmark"
    }
  ],
  },
  "links": [
    {
      "href": "http://openstack.example.com/v2/openstack/servers/
dd3b0715-a3fc-43d8-bbd2-2720beb226fb",
      "rel": "self"
    },
    {
      "href": "http://openstack.example.com/openstack/servers/
dd3b0715-a3fc-43d8-bbd2-2720beb226fb",
      "rel": "bookmark"
    }
  ],
  "metadata": {
    "My Server Name": "Apache1"
  },
  "name": "new-server-test",
  "progress": 0,
  "status": "ACTIVE",
  "tenant_id": "openstack",
  "updated": "2013-02-04T13:17:51Z",
  "user_id": "fake"
}
}

```

Example 3.393. Get server information with configuration drive: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1" status="ACTIVE" updated="2013-02-04T13:26:10Z"
  hostId="7a8c3fc15db5d6227d26d5ef559b77c880bbe99da5ce5f5871fc113e"
  name="new-server-test" created="2013-02-04T13:26:09Z" userId="fake"
  tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id=
"3b9e0572-3d7b-4e6f-9c21-35ad0f7dbf95" config_drive="">
  <image id="70a599e0-31e7-49b7-b260-868f441e862b">
    <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  </image>
  <flavor id="1">
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="private">
      <ip version="4" addr="192.168.0.3"/>
    </network>
  </addresses>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/
3b9e0572-3d7b-4e6f-9c21-35ad0f7dbf95" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/
3b9e0572-3d7b-4e6f-9c21-35ad0f7dbf95" rel="bookmark"/>

```

```
</server>
```

This operation does not return a response body.

3.46.3. Get server details with configuration drive

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/detail	Lists details for all servers including the configuration drive extended attribute.

Normal response codes: 202

3.46.3.1. Request

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

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

This operation does not require a request body.

3.46.3.2. Response

Example 3.394. Get server details with configuration drive: JSON response

```
{
  "servers": [
    {
      "accessIPv4": "",
      "accessIPv6": "",
      "addresses": {
        "private": [
          {
            "addr": "192.168.0.3",
            "version": 4
          }
        ]
      },
      "config_drive": "",
      "created": "2013-02-04T13:21:44Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "hostId":
"76e154b0015e25fad65a7ab0c35a86dd79acfa8312075a6534ef6176",
      "id": "720e688f-5ec8-4d4f-b585-dbd1a89ceeb0",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
```

```

        {
            "href": "http://openstack.example.com/openstack/
images/70a599e0-31e7-49b7-b260-868f441e862b",
            "rel": "bookmark"
        }
    ],
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/
servers/720e688f-5ec8-4d4f-b585-dbd1a89ceeb0",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/servers/
720e688f-5ec8-4d4f-b585-dbd1a89ceeb0",
            "rel": "bookmark"
        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2013-02-04T13:21:44Z",
    "user_id": "fake"
}
]
}

```

Example 3.395. Get server details with configuration drive: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
  <server status="ACTIVE" updated="2013-02-04T13:26:27Z" hostId=
"2a00edcff768661880eb9c96c951f56c2c5dcd873bb652361008efc7" name="new-
server-test" created="2013-02-04T13:26:27Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="515d94d3-ae4-4bd5-
bb4e-9601c657372f" config_drive="">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
      <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
      <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
      <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
      <network id="private">
        <ip version="4" addr="192.168.0.3"/>
      </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/
515d94d3-ae4-4bd5-bb4e-9601c657372f" rel="self"/>
  </server>
</servers>

```

```
<atom:link href="http://openstack.example.com/openstack/servers/515d94d3-
aee4-4bd5-bb4e-9601c657372f" rel="bookmark" />
</server>
</servers>
```

This operation does not return a response body.

3.47. Servers with extended availability zones (servers)

Show the instance availability zone for compute nodes (nova-compute). Internal services appear in their own *internal* availability zone.

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server, including its availability zone.
GET	/v2/{tenant_id}/servers/detail	Lists details for servers, including their current availability zone.

3.47.1. Show server

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server, including its availability zone.

Normal response codes: 200

3.47.1.1. Request

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

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

This operation does not require a request body.

3.47.1.2. Response

Example 3.396. Show server: JSON response

```
{
  "server": {
    "OS-EXT-AZ:availability_zone": "nova",
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "created": "2013-01-30T13:38:47Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "d38ea49a033b0efaf80c165de63f4805c886dfb94dc0fe731227eccb",
    "id": "fb7babfd-e1a1-4add-90e6-3558180983c7",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    }
  }
}
```



```

    ],
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/
fb7babfd-e1a1-4add-90e6-3558180983c7",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/
fb7babfd-e1a1-4add-90e6-3558180983c7",
        "rel": "bookmark"
      }
    ],
    "metadata": {
      "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2013-01-30T13:38:49Z",
    "user_id": "fake"
  }
}

```

Example 3.397. Show server: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-EXT-AZ="http://docs.openstack.org/compute/ext/
extended_availability_zone/api/v2" xmlns:atom="http://www.w3.
org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/
v1.1" status="ACTIVE" updated="2013-01-30T14:29:20Z" hostId=
"471e52951e3182954c5a93489dafc3fc38a9ef3e0b62d26dc740460c" name="new-
server-test" created="2013-01-30T14:29:19Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="26ea8424-758d-483a-
addc-9a5905afc9e6" OS-EXT-AZ:availability_zone="nova">
  <image id="70a599e0-31e7-49b7-b260-868f441e862b">
    <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  </image>
  <flavor id="1">
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="private">
      <ip version="4" addr="192.168.0.3"/>
    </network>
  </addresses>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/
26ea8424-758d-483a-addc-9a5905afc9e6" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/
26ea8424-758d-483a-addc-9a5905afc9e6" rel="bookmark"/>
</server>

```

This operation does not return a response body.

3.47.2. List details for servers

Method	URI	Description
GET	/v2/{tenant_id}/servers/detail	Lists details for servers, including their current availability zone.

Normal response codes: 200

3.47.2.1. Request

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

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

This operation does not require a request body.

3.47.2.2. Response

Example 3.398. List details for servers: JSON response

```
{
  "servers": [
    {
      "OS-EXT-AZ:availability_zone": "nova",
      "accessIPv4": "",
      "accessIPv6": "",
      "addresses": {
        "private": [
          {
            "addr": "192.168.0.3",
            "version": 4
          }
        ]
      },
      "created": "2013-01-30T13:26:51Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "hostId":
"60c988a84401fa15888a32833e5848e9caa99a45778310ba7b363165",
      "id": "3dbf5b00-dabc-41ff-b6ab-4409568fae9d",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
images/70a599e0-31e7-49b7-b260-868f441e862b",

```

```

        "rel": "bookmark"
      }
    ]
  },
  "links": [
    {
      "href": "http://openstack.example.com/v2/openstack/
servers/3dbf5b00-dabc-41ff-b6ab-4409568fae9d",
      "rel": "self"
    },
    {
      "href": "http://openstack.example.com/openstack/servers/
3dbf5b00-dabc-41ff-b6ab-4409568fae9d",
      "rel": "bookmark"
    }
  ],
  "metadata": {
    "My Server Name": "Apache1"
  },
  "name": "new-server-test",
  "progress": 0,
  "status": "ACTIVE",
  "tenant_id": "openstack",
  "updated": "2013-01-30T13:26:52Z",
  "user_id": "fake"
}
]
}

```

Example 3.399. List details for servers: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:OS-EXT-AZ="http://docs.openstack.org/compute/ext/
extended_availability_zone/api/v2" xmlns:atom="http://www.w3.org/2005/Atom"
xmlns="http://docs.openstack.org/compute/api/v1.1">
  <server status="ACTIVE" updated="2013-01-30T14:29:20Z" hostId=
"85adf7d0492dedf0a7e3dc44ef7d16186b768ca3df33c4d608e630d9" name="new-server-
test" created="2013-01-30T14:29:19Z" userId="fake" tenantId="openstack"
accessIPv4="" accessIPv6="" progress="0" id="a668c72d-2bac-4806-a297-
c7c11d97e3b3" OS-EXT-AZ:availability_zone="nova">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
      <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
      <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
      <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
      <network id="private">
        <ip version="4" addr="192.168.0.3"/>
      </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/
a668c72d-2bac-4806-a297-c7c11d97e3b3" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/
a668c72d-2bac-4806-a297-c7c11d97e3b3" rel="bookmark"/>
  </server>
</servers>

```

```
</server>
</servers>
```

This operation does not return a response body.

3.48. Servers and images with disk config (servers, images)

Extend servers with the diskConfig attribute.

Method	URI	Description
POST	/v2/{tenant_id}/servers	Creates a server.
GET	/v2/{tenant_id}/servers/{server_id}	Shows information for a specified server.
PUT	/v2/{tenant_id}/servers/{server_id}	Updates a specified server.
POST	/v2/{tenant_id}/servers/{server_id}/action	Resizes a server.
POST	/v2/{tenant_id}/servers/{server_id}/action	Rebuilds a specified server.
GET	/v2/{tenant_id}/servers/detail	Lists servers.
GET	/v2/{tenant_id}/images/{image_id}	Gets information for a specified image.
GET	/v2/{tenant_id}/images/detail	Lists images.


```

        b25zLiINCgOKLVJpY2hhcmQgQmFjaA==
    </file>
  </personality>
</server>

```

This operation does not require a request body.

3.48.1.2. Response

Example 3.402. Create server: JSON response

```

{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "adminPass": "CQH9gWzgkVno",
    "id": "324dfb7d-f4a9-419a-9a19-237df04b443b",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/324dfb7d-f4a9-419a-9a19-237df04b443b",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/324dfb7d-f4a9-419a-9a19-237df04b443b",
        "rel": "bookmark"
      }
    ]
  }
}

```

Example 3.403. Create server: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" id="900a4ef7-f374-413f-8816-52d3dbfaf498" adminPass="Cj6sZgWq85qm" OS-DCF:diskConfig="AUTO">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/900a4ef7-f374-413f-8816-52d3dbfaf498" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/900a4ef7-f374-413f-8816-52d3dbfaf498" rel="bookmark"/>
</server>

```

This operation does not return a response body.

3.48.2. Show server information

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

Normal response codes: 200

3.48.2.1. Request

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

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

This operation does not require a request body.

3.48.2.2. Response

Example 3.404. Show server information: JSON response

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "created": "2012-12-02T02:11:55Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "c949ab4256cea23b6089b710aa2df48bf6577ed915278b62e33ad8bb",
    "id": "5046e2f2-3b33-4041-b3cf-e085f73e78e7",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    }
  }
}
```

```

    }
  ],
  "links": [
    {
      "href": "http://openstack.example.com/v2/openstack/servers/5046e2f2-3b33-4041-b3cf-e085f73e78e7",
      "rel": "self"
    },
    {
      "href": "http://openstack.example.com/openstack/servers/5046e2f2-3b33-4041-b3cf-e085f73e78e7",
      "rel": "bookmark"
    }
  ],
  "metadata": {
    "My Server Name": "Apache1"
  },
  "name": "new-server-test",
  "progress": 0,
  "status": "ACTIVE",
  "tenant_id": "openstack",
  "updated": "2012-12-02T02:11:55Z",
  "user_id": "fake"
}
}

```

Example 3.405. Show server information: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/
api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1" status="ACTIVE" updated="2012-12-02T02:15:37Z"
  hostId="afa0a883de4743c7a0c164327bda5284b875c50e1a9e30de910ac126" name=
"new-server-test" created="2012-12-02T02:15:37Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="d5d844c0-
ecc3-4202-90ed-2e85b7fe513d" OS-DCF:diskConfig="AUTO">
  <image id="70a599e0-31e7-49b7-b260-868f441e862b">
    <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  </image>
  <flavor id="1">
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="private">
      <ip version="4" addr="192.168.0.3"/>
    </network>
  </addresses>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/d5d844c0-
ecc3-4202-90ed-2e85b7fe513d" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/d5d844c0-
ecc3-4202-90ed-2e85b7fe513d" rel="bookmark"/>
</server>

```

This operation does not return a response body.

3.48.3. Update server

Method	URI	Description
PUT	/v2/{tenant_id}/servers/{server_id}	Updates a specified server.

Normal response codes: 200

3.48.3.1. Request

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

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

Example 3.406. Update server: JSON request

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

Example 3.407. Update server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  OS-DCF:diskConfig="AUTO" />
```

This operation does not require a request body.

3.48.3.2. Response

Example 3.408. Update server: JSON response

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "created": "2012-12-02T02:11:57Z",
    "flavor": {
```

```

        "id": "1",
        "links": [
            {
                "href": "http://openstack.example.com/openstack/flavors/
1",
                "rel": "bookmark"
            }
        ]
    },
    "hostId": "6e84af987b4e7ec1c039b16d21f508f4a505672bd94fb0218b668d07",
    "id": "324dfb7d-f4a9-419a-9a19-237df04b443b",
    "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
            {
                "href": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
                "rel": "bookmark"
            }
        ]
    },
    "links": [
        {
            "href": "http://openstack.example.com/v2/openstack/servers/
324dfb7d-f4a9-419a-9a19-237df04b443b",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/servers/
324dfb7d-f4a9-419a-9a19-237df04b443b",
            "rel": "bookmark"
        }
    ],
    "metadata": {
        "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2012-12-02T02:11:58Z",
    "user_id": "fake"
}
}

```

Example 3.409. Update server: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/
api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1" status="ACTIVE" updated="2012-12-02T02:15:40Z"
  hostId="e987bebf62599c59c4559b249d0f5f300a302d1e9ff22dd1f0c5c4b0" name=
"new-server-test" created="2012-12-02T02:15:40Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="900a4ef7-
f374-413f-8816-52d3dbfaf498" OS-DCF:diskConfig="AUTO">
  <image id="70a599e0-31e7-49b7-b260-868f441e862b">
    <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  </image>
</flavor id="1">

```

```
<atom:link href="http://openstack.example.com/openstack/flavors/1" rel="bookmark"/>
</flavor>
<metadata>
  <meta key="My Server Name">Apache1</meta>
</metadata>
<addresses>
  <network id="private">
    <ip version="4" addr="192.168.0.3"/>
  </network>
</addresses>
<atom:link href="http://openstack.example.com/v2/openstack/servers/900a4ef7-f374-413f-8816-52d3dbfaf498" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/servers/900a4ef7-f374-413f-8816-52d3dbfaf498" rel="bookmark"/>
</server>
```

This operation does not return a response body.

3.48.4. Resize server

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

Normal response codes: 202

3.48.4.1. Request

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

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

Example 3.410. Resize server: JSON request

```
{
  "resize": {
    "flavorRef": "3",
    "OS-DCF:diskConfig": "AUTO"
  }
}
```

Example 3.411. Resize server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<resize xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  OS-DCF:diskConfig="AUTO"
  flavorRef="3" />
```

This operation does not require a request body.

3.48.5. Rebuild server

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

Normal response codes: 202

3.48.5.1. Request

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

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

Example 3.412. Rebuild server: JSON request

```
{
  "rebuild": {
    "imageRef" : "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
    "OS-DCF:diskConfig": "AUTO"
  }
}
```

Example 3.413. Rebuild server: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<rebuild xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/v1.1"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  imageRef="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b"
  OS-DCF:diskConfig="AUTO" />
```

This operation does not require a request body.

3.48.5.2. Response

Example 3.414. Rebuild server: JSON response

```
{
  "server": {
    "OS-DCF:diskConfig": "AUTO",
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    }
  }
}
```

```

    ],
    "adminPass": "NBjMaJoFL4EF",
    "created": "2012-12-02T02:11:56Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/
1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "c076393ad900d62c4805a42df10d9b364f629842681c00cce035487f",
    "id": "63a8aa13-60fe-41c4-b079-77f6fdf3c841",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    },
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/
63a8aa13-60fe-41c4-b079-77f6fdf3c841",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/
63a8aa13-60fe-41c4-b079-77f6fdf3c841",
        "rel": "bookmark"
      }
    ],
    "metadata": {
      "My Server Name": "Apache1"
    },
    "name": "new-server-test",
    "progress": 0,
    "status": "ACTIVE",
    "tenant_id": "openstack",
    "updated": "2012-12-02T02:11:56Z",
    "user_id": "fake"
  }
}

```

Example 3.415. Rebuild server: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/
api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1" status="ACTIVE" updated="2012-12-02T02:15:39Z"
hostId="981de784ae4d8c49ca075024977828a16e7f3c2beeb19115b0366e17" name=
"new-server-test" created="2012-12-02T02:15:38Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="10791a94-8900-4d0c-
b93d-0debb224882e" adminPass="mTxoVD3eALpv" OS-DCF:diskConfig="AUTO">

```

```
<image id="70a599e0-31e7-49b7-b260-868f441e862b">
  <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
</image>
<flavor id="1">
  <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
</flavor>
<metadata>
  <meta key="My Server Name">Apache1</meta>
</metadata>
<addresses>
  <network id="private">
    <ip version="4" addr="192.168.0.3"/>
  </network>
</addresses>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/
10791a94-8900-4d0c-b93d-0debb224882e" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/
10791a94-8900-4d0c-b93d-0debb224882e" rel="bookmark"/>
</server>
```

This operation does not return a response body.

3.48.6. List servers

Method	URI	Description
GET	/v2/{tenant_id}/servers/detail	Lists servers.

Normal response codes: 200

3.48.6.1. Request

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

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

This operation does not require a request body.

3.48.6.2. Response

Example 3.416. List servers: JSON response

```
{
  "servers": [
    {
      "OS-DCF:diskConfig": "AUTO",
      "accessIPv4": "",
      "accessIPv6": "",
      "addresses": {
        "private": [
          {
            "addr": "192.168.0.3",
            "version": 4
          }
        ]
      },
      "created": "2012-12-02T02:11:55Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "hostId":
"99428f32351a5d89d0f7727c6eec68c1777c545a0972aac645508dc",
      "id": "05372e62-05b9-4ee2-9343-9a1fdf2a5fda",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
images/70a599e0-31e7-49b7-b260-868f441e862b",
            "rel": "bookmark"
          }
        ]
      }
    }
  ]
}
```



```

    }
  ],
  "links": [
    {
      "href": "http://openstack.example.com/v2/openstack/
servers/05372e62-05b9-4ee2-9343-9a1fdf2a5fda",
      "rel": "self"
    },
    {
      "href": "http://openstack.example.com/openstack/servers/
05372e62-05b9-4ee2-9343-9a1fdf2a5fda",
      "rel": "bookmark"
    }
  ],
  "metadata": {
    "My Server Name": "Apache1"
  },
  "name": "new-server-test",
  "progress": 0,
  "status": "ACTIVE",
  "tenant_id": "openstack",
  "updated": "2012-12-02T02:11:56Z",
  "user_id": "fake"
}
]
}

```

Example 3.417. List servers: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/
v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1">
  <server status="ACTIVE" updated="2012-12-02T02:15:38Z" hostId=
"85973b57730e91f4eea36b3e7a2a7e3fdaf56008af335dd59f897a59" name=
"new-server-test" created="2012-12-02T02:15:38Z" userId="fake"
tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id=
"08266bed-2651-4b6c-9dc8-83f0c3ef9d38" OS-DCF:diskConfig="AUTO">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
      <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
      <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
      <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
      <network id="private">
        <ip version="4" addr="192.168.0.3"/>
      </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/
08266bed-2651-4b6c-9dc8-83f0c3ef9d38" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/
08266bed-2651-4b6c-9dc8-83f0c3ef9d38" rel="bookmark"/>
  </server>

```

```
</servers>
```

This operation does not return a response body.

3.48.7. Get image information

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

Normal response codes: 200

3.48.7.1. Request

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

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

This operation does not require a request body.

3.48.7.2. Response

Example 3.418. Get image information: JSON response

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

```
    "updated": "2011-01-01T01:02:03Z"  
  }  
}
```

Example 3.419. Get image information: XML response

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

This operation does not return a response body.

3.48.8. List images

Method	URI	Description
GET	/v2/{tenant_id}/images/detail	Lists images.

Normal response codes: 200

3.48.8.1. Request

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

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

This operation does not require a request body.

3.48.8.2. Response

Example 3.420. List images: JSON response

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

```

    {
      "created": "2011-01-01T01:02:03Z",
      "id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
          "rel": "bookmark"
        },
        {
          "href": "http://glance.openstack.example.com/openstack/images/155d900f-4e14-4e4c-a73d-069cbf4541e6",
          "rel": "alternate",
          "type": "application/vnd.openstack.image"
        }
      ],
      "metadata": {
        "architecture": "x86_64",
        "kernel_id": "nokernel",
        "ramdisk_id": "nokernel"
      },
      "minDisk": 0,
      "minRam": 0,
      "name": "fakeimage123456",
      "progress": 100,
      "status": "ACTIVE",
      "updated": "2011-01-01T01:02:03Z"
    },
    {
      "created": "2011-01-01T01:02:03Z",
      "id": "a2459075-d96c-40d5-893e-577ff92e721c",
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",
          "rel": "bookmark"
        },
        {
          "href": "http://glance.openstack.example.com/openstack/images/a2459075-d96c-40d5-893e-577ff92e721c",
          "rel": "alternate",
          "type": "application/vnd.openstack.image"
        }
      ],
      "metadata": {
        "kernel_id": "nokernel",
        "ramdisk_id": "nokernel"
      },
      "minDisk": 0,
      "minRam": 0,

```

```

        "name": "fakeimage123456",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    },
    {
        "OS-DCF:diskConfig": "MANUAL",
        "created": "2011-01-01T01:02:03Z",
        "id": "a440c04b-79fa-479c-bed1-0b816eae379",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/images/a440c04b-79fa-479c-bed1-0b816eae379",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eae379",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/images/a440c04b-79fa-479c-bed1-0b816eae379",
                "rel": "alternate",
                "type": "application/vnd.openstack.image"
            }
        ],
        "metadata": {
            "architecture": "x86_64",
            "auto_disk_config": "False",
            "kernel_id": "nokernel",
            "ramdisk_id": "nokernel"
        },
        "minDisk": 0,
        "minRam": 0,
        "name": "fakeimage6",
        "progress": 100,
        "status": "ACTIVE",
        "updated": "2011-01-01T01:02:03Z"
    },
    {
        "created": "2011-01-01T01:02:03Z",
        "id": "c905cedb-7281-47e4-8a62-f26bc5fc4c77",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "bookmark"
            },
            {
                "href": "http://glance.openstack.example.com/openstack/images/c905cedb-7281-47e4-8a62-f26bc5fc4c77",
                "rel": "alternate",
                "type": "application/vnd.openstack.image"
            }
        ]
    }

```

```

    ],
    "metadata": {
      "kernel_id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
      "ramdisk_id": null
    },
    "minDisk": 0,
    "minRam": 0,
    "name": "fakeimage123456",
    "progress": 100,
    "status": "ACTIVE",
    "updated": "2011-01-01T01:02:03Z"
  },
  {
    "created": "2011-01-01T01:02:03Z",
    "id": "cedef40a-ed67-4d10-800e-17455edce175",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175",
        "rel": "bookmark"
      },
      {
        "href": "http://glance.openstack.example.com/openstack/
images/cedef40a-ed67-4d10-800e-17455edce175",
        "rel": "alternate",
        "type": "application/vnd.openstack.image"
      }
    ],
    "metadata": {
      "kernel_id": "nokernel",
      "ramdisk_id": "nokernel"
    },
    "minDisk": 0,
    "minRam": 0,
    "name": "fakeimage123456",
    "progress": 100,
    "status": "ACTIVE",
    "updated": "2011-01-01T01:02:03Z"
  },
  {
    "created": "2011-01-01T01:02:03Z",
    "id": "76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
        "rel": "bookmark"
      }
    ],
  }

```



```

        "href": "http://glance.openstack.example.com/openstack/
images/76fa36fc-c930-4bf3-8c8a-ea2a2420deb6",
        "rel": "alternate",
        "type": "application/vnd.openstack.image"
    }
],
"metadata": {
    "kernel_id": "nokernel",
    "ramdisk_id": "nokernel"
},
"minDisk": 0,
"minRam": 0,
"name": "fakeimage123456",
"progress": 100,
"status": "ACTIVE",
"updated": "2011-01-01T01:02:03Z"
}
]
}

```

Example 3.421. List images: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<images xmlns:OS-DCF="http://docs.openstack.org/compute/ext/disk_config/api/
v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1">
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage7"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"70a599e0-31e7-49b7-b260-868f441e862b" OS-DCF:diskConfig="AUTO">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="auto_disk_config">True</meta>
      <meta key="ramdisk_id">nokernel</meta>
      <meta key="architecture">x86_64</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"155d900f-4e14-4e4c-a73d-069cbf4541e6">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="ramdisk_id">nokernel</meta>
      <meta key="architecture">x86_64</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
155d900f-4e14-4e4c-a73d-069cbf4541e6" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>

```

```

<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"a2459075-d96c-40d5-893e-577ff92e721c">
  <metadata>
    <meta key="kernel_id">nokernel</meta>
    <meta key="ramdisk_id">nokernel</meta>
  </metadata>
  <atom:link href="http://openstack.example.com/v2/openstack/images/
a2459075-d96c-40d5-893e-577ff92e721c" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/images/a2459075-
d96c-40d5-893e-577ff92e721c" rel="bookmark"/>
  <atom:link href="http://glance.openstack.example.com/openstack/images/
a2459075-d96c-40d5-893e-577ff92e721c" type="application/vnd.openstack.image"
rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage6"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"a440c04b-79fa-479c-bed1-0b816eaec379" OS-DCF:diskConfig="MANUAL">
  <metadata>
    <meta key="kernel_id">nokernel</meta>
    <meta key="auto_disk_config">False</meta>
    <meta key="ramdisk_id">nokernel</meta>
    <meta key="architecture">x86_64</meta>
  </metadata>
  <atom:link href="http://openstack.example.com/v2/openstack/images/
a440c04b-79fa-479c-bed1-0b816eaec379" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/images/
a440c04b-79fa-479c-bed1-0b816eaec379" rel="bookmark"/>
  <atom:link href="http://glance.openstack.example.com/openstack/images/
a440c04b-79fa-479c-bed1-0b816eaec379" type="application/vnd.openstack.image"
rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"c905cedb-7281-47e4-8a62-f26bc5fc4c77">
  <metadata>
    <meta key="kernel_id">155d900f-4e14-4e4c-a73d-069cbf4541e6</meta>
    <meta key="ramdisk_id">None</meta>
  </metadata>
  <atom:link href="http://openstack.example.com/v2/openstack/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77" rel="bookmark"/>
  <atom:link href="http://glance.openstack.example.com/openstack/images/
c905cedb-7281-47e4-8a62-f26bc5fc4c77" type="application/vnd.openstack.image"
rel="alternate"/>
</image>
<image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"cedef40a-ed67-4d10-800e-17455edce175">
  <metadata>
    <meta key="kernel_id">nokernel</meta>
    <meta key="ramdisk_id">nokernel</meta>
  </metadata>
  <atom:link href="http://openstack.example.com/v2/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/images/cedef40a-
ed67-4d10-800e-17455edce175" rel="bookmark"/>

```

```

    <atom:link href="http://glance.openstack.example.com/openstack/images/
cedef40a-ed67-4d10-800e-17455edce175" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
  <image status="ACTIVE" updated="2011-01-01T01:02:03Z" name="fakeimage123456"
created="2011-01-01T01:02:03Z" minDisk="0" progress="100" minRam="0" id=
"76fa36fc-c930-4bf3-8c8a-ea2a2420deb6">
    <metadata>
      <meta key="kernel_id">nokernel</meta>
      <meta key="ramdisk_id">nokernel</meta>
    </metadata>
    <atom:link href="http://openstack.example.com/v2/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/images/76fa36fc-
c930-4bf3-8c8a-ea2a2420deb6" rel="bookmark"/>
    <atom:link href="http://glance.openstack.example.com/openstack/images/
76fa36fc-c930-4bf3-8c8a-ea2a2420deb6" type="application/vnd.openstack.image"
rel="alternate"/>
  </image>
</images>

```

This operation does not return a response body.

3.49. Server IP type (servers)

Show the type of the IP addresses assigned to an instance. Type is either fixed or floating.

Method	URI	Description
GET	/v2/{tenant_id}/servers/ {server_id}/action	Shows the type of IP assigned to a specified server, either fixed or floating.
GET	/v2/{tenant_id}/servers/detail	Lists all servers showing their IPs by type, either fixed or floating.

3.49.1. Show IP type

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}/action	Shows the type of IP assigned to a specified server, either fixed or floating.

Normal response codes: 200

3.49.1.1. Request

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

Name	Type	Description
{tenant_id}	String	The unique identifier of the tenant or account.
{server_id}	UUID	The UUID for the server of interest to you.

This operation does not require a request body.

3.49.1.2. Response

Example 3.422. Show IP type: JSON response

```
{
  "server": {
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "OS-EXT-IPS:type": "fixed",
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "created": "2013-02-07T18:46:28Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "4e2003eddbfdb1280c2618d04090bcdd6773203b8da8347af0b2723d",
    "id": "dc7281f9-ee47-40b9-9950-9f73e7961caa",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    }
  }
}
```

```

    }
  ],
  "links": [
    {
      "href": "http://openstack.example.com/v2/openstack/servers/
dc7281f9-ee47-40b9-9950-9f73e7961caa",
      "rel": "self"
    },
    {
      "href": "http://openstack.example.com/openstack/servers/
dc7281f9-ee47-40b9-9950-9f73e7961caa",
      "rel": "bookmark"
    }
  ],
  "metadata": {
    "My Server Name": "Apache1"
  },
  "name": "new-server-test",
  "progress": 0,
  "status": "ACTIVE",
  "tenant_id": "openstack",
  "updated": "2013-02-07T18:46:29Z",
  "user_id": "fake"
}
}

```

Example 3.423. Show IP type: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-EXT-IPS="http://docs.openstack.org/compute/ext/extended_ips/
api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1" status="ACTIVE" updated="2013-02-07T18:46:29Z"
  hostId="068cc5e2de14b6e533a239c6eac0a0bdecd57cab25450a6d3da43af" name=
"new-server-test" created="2013-02-07T18:46:28Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="22e7cf4d-
ab7a-4a3d-9599-7d0dbaf9ed55">
  <image id="70a599e0-31e7-49b7-b260-868f441e862b">
    <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  </image>
  <flavor id="1">
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="private">
      <ip OS-EXT-IPS:type="fixed" version="4" addr="192.168.0.3"/>
    </network>
  </addresses>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/22e7cf4d-
ab7a-4a3d-9599-7d0dbaf9ed55" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/22e7cf4d-
ab7a-4a3d-9599-7d0dbaf9ed55" rel="bookmark"/>
</server>

```

This operation does not return a response body.

3.49.2. List servers with IP type

Method	URI	Description
GET	/v2/{tenant_id}/servers/detail	Lists all servers showing their IPs by type, either fixed or floating.

Normal response codes: 200

3.49.2.1. Request

This table shows the URI parameters for the list servers with ip type request:

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

This operation does not require a request body.

3.49.2.2. Response

Example 3.424. List servers with IP type: JSON response

```
{
  "servers": [
    {
      "accessIPv4": "",
      "accessIPv6": "",
      "addresses": {
        "private": [
          {
            "OS-EXT-IPS:type": "fixed",
            "addr": "192.168.0.3",
            "version": 4
          }
        ]
      },
      "created": "2013-02-07T18:40:59Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "hostId":
"fe866a4962fe3bdb6c2db9c8f7dcdb955aca73387e72b5cb9c45bd3",
      "id": "76908712-653a-4d16-807e-d89d41435d24",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
images/70a599e0-31e7-49b7-b260-868f441e862b",
```

```

        "rel": "bookmark"
      }
    ]
  },
  "links": [
    {
      "href": "http://openstack.example.com/v2/openstack/
servers/76908712-653a-4d16-807e-d89d41435d24",
      "rel": "self"
    },
    {
      "href": "http://openstack.example.com/openstack/servers/
76908712-653a-4d16-807e-d89d41435d24",
      "rel": "bookmark"
    }
  ],
  "metadata": {
    "My Server Name": "Apache1"
  },
  "name": "new-server-test",
  "progress": 0,
  "status": "ACTIVE",
  "tenant_id": "openstack",
  "updated": "2013-02-07T18:40:59Z",
  "user_id": "fake"
}
]
}

```

Example 3.425. List servers with IP type: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:OS-EXT-IPS="http://docs.openstack.org/compute/ext/extended_ips/
api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.
openstack.org/compute/api/v1.1">
  <server status="ACTIVE" updated="2013-02-07T18:40:59Z" hostId=
"51a80e6ee89b638b2cb57eb4e39d89a725e07c8a698f4d8e256f8665" name=
"new-server-test" created="2013-02-07T18:40:59Z" userId="fake"
tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id=
"0337de6b-1d43-46c8-8804-35669f1dea9a">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
      <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
      <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
      <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
      <network id="private">
        <ip OS-EXT-IPS:type="fixed" version="4" addr="192.168.0.3"/>
      </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/
0337de6b-1d43-46c8-8804-35669f1dea9a" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/
0337de6b-1d43-46c8-8804-35669f1dea9a" rel="bookmark"/>
  </server>
</servers>

```

```
</server>  
</servers>
```

This operation does not return a response body.

3.50. Server extended attributes (servers)

Show metadata for servers.

Method	URI	Description
GET	/v2/servers	Lists detailed extended server attribute information for all servers.
GET	/v2/servers/{server_id}	Shows extended server attributes for a specified server.

3.50.1. List servers with extended server attributes

Method	URI	Description
GET	/v2/servers	Lists detailed extended server attribute information for all servers.

Normal response codes: 200

3.50.1.1. Request

This operation does not require a request body.

3.50.1.2. Response

Example 3.426. List servers with extended server attributes: JSON response

```
{
  "servers": [
    {
      "OS-EXT-SRV-ATTR:host": "dd99797793774612b081a8be19bf721a",
      "OS-EXT-SRV-ATTR:hypervisor_hostname": "fake-mini",
      "OS-EXT-SRV-ATTR:instance_name": "instance-00000001",
      "accessIPv4": "",
      "accessIPv6": "",
      "addresses": {
        "private": [
          {
            "addr": "192.168.0.3",
            "version": 4
          }
        ]
      },
      "created": "2012-11-15T19:27:05Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "hostId":
"146245c049213a54b8c2352751518fcb4c2befd1b942b45a5a705d35",
      "id": "e0c3563a-84ef-4d0b-bb80-23392cd23882",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
images/70a599e0-31e7-49b7-b260-868f441e862b",
            "rel": "bookmark"
          }
        ]
      }
    }
  ],
}
```

```

        "links": [
          {
            "href": "http://openstack.example.com/v2/openstack/
servers/e0c3563a-84ef-4d0b-bb80-23392cd23882",
            "rel": "self"
          },
          {
            "href": "http://openstack.example.com/openstack/servers/
e0c3563a-84ef-4d0b-bb80-23392cd23882",
            "rel": "bookmark"
          }
        ],
        "metadata": {
          "My Server Name": "Apache1"
        },
        "name": "new-server-test",
        "progress": 0,
        "status": "ACTIVE",
        "tenant_id": "openstack",
        "updated": "2012-11-15T19:27:05Z",
        "user_id": "fake"
      }
    ]
  }
}

```

Example 3.427. List servers with extended server attributes: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:OS-EXT-SRV-ATTR="http://docs.openstack.org/compute/ext/
extended_status/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1">
  <server status="ACTIVE" updated="2012-11-15T19:27:06Z" hostId=
"b348a7376e2e61781829c9b45e63675aa0207632c25ce36c55a4fb2a" name="new-
server-test" created="2012-11-15T19:27:06Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="3cadb9e9-
f430-4f62-8b9b-3efb671ff1fa" OS-EXT-SRV-ATTR:instance_name="instance-00000001"
OS-EXT-SRV-ATTR:host="2c4d049170fe409abc14942757d63a4e" OS-EXT-SRV-
ATTR:hypervisor_hostname="fake-mini">
    <image id="70a599e0-31e7-49b7-b260-868f441e862b">
      <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
    </image>
    <flavor id="1">
      <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
    </flavor>
    <metadata>
      <meta key="My Server Name">Apache1</meta>
    </metadata>
    <addresses>
      <network id="private">
        <ip version="4" addr="192.168.0.3"/>
      </network>
    </addresses>
    <atom:link href="http://openstack.example.com/v2/openstack/servers/
3cadb9e9-f430-4f62-8b9b-3efb671ff1fa" rel="self"/>
    <atom:link href="http://openstack.example.com/openstack/servers/3cadb9e9-
f430-4f62-8b9b-3efb671ff1fa" rel="bookmark"/>
  </server>
</servers>

```

This operation does not return a response body.

3.50.2. Show extended server attributes

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

Normal response codes: 200

3.50.2.1. Request

This table shows the URI parameters for the show extended server attributes request:

Name	Type	Description
{server_id}	UUID	The UUID for the server of interest to you.

This operation does not require a request body.

3.50.2.2. Response

Example 3.428. Show extended server attributes: JSON response

```
{
  "server": {
    "OS-EXT-SRV-ATTR:host": "1169a68456af48238da47b1d5957a714",
    "OS-EXT-SRV-ATTR:hypervisor_hostname": "fake-mini",
    "OS-EXT-SRV-ATTR:instance_name": "instance-00000001",
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "created": "2012-11-15T19:27:04Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "2dfce43c41dd288cfac3a5b4251742b3bd2b37c12eb5927e757d9b4c",
    "id": "1fc2392e-5727-46af-bc21-317a4a3eb04c",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
          "rel": "bookmark"
        }
      ]
    }
  }
}
```

```

    }
  ],
  "links": [
    {
      "href": "http://openstack.example.com/v2/openstack/servers/1fc2392e-5727-46af-bc21-317a4a3eb04c",
      "rel": "self"
    },
    {
      "href": "http://openstack.example.com/openstack/servers/1fc2392e-5727-46af-bc21-317a4a3eb04c",
      "rel": "bookmark"
    }
  ],
  "metadata": {
    "My Server Name": "Apache1"
  },
  "name": "new-server-test",
  "progress": 0,
  "status": "ACTIVE",
  "tenant_id": "openstack",
  "updated": "2012-11-15T19:27:04Z",
  "user_id": "fake"
}
}

```

Example 3.429. Show extended server attributes: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-EXT-SRV-ATTR="http://docs.openstack.org/compute/ext/extended_status/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" status="ACTIVE" updated="2012-11-15T19:27:06Z" hostId="6b8205d183f40afad106dbeac44d3872151ef6f36790077ea2ea85fc" name="new-server-test" created="2012-11-15T19:27:05Z" userId="fake" tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id="ece641c1-51f5-4190-9342-d9751f28eead" OS-EXT-SRV-ATTR:instance_name="instance-00000001" OS-EXT-SRV-ATTR:host="80edfa5af48b4894b20eb1d9d2d4424e" OS-EXT-SRV-ATTR:hypervisor_hostname="fake-mini">
  <image id="70a599e0-31e7-49b7-b260-868f441e862b">
    <atom:link href="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  </image>
  <flavor id="1">
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel="bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="private">
      <ip version="4" addr="192.168.0.3"/>
    </network>
  </addresses>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/ece641c1-51f5-4190-9342-d9751f28eead" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/ece641c1-51f5-4190-9342-d9751f28eead" rel="bookmark"/>
</server>

```

```
</server>
```

This operation does not return a response body.

3.51. Server extended status (servers)

Show extended status information, `vm_state`, `task_state`, and `power_state`, in detailed server responses.

Method	URI	Description
GET	<code>/v2/{tenant_id}/servers/{server_id}</code>	Shows the extended status attributes in the response for a specified server.
GET	<code>/v2/{tenant_id}/servers/detail</code>	Lists the extended status attributes in the detailed response for all servers.

3.51.1. Show server extended status

Method	URI	Description
GET	/v2/{tenant_id}/servers/{server_id}	Shows the extended status attributes in the response for a specified server.

The extended status attributes are `vm_state`, `power_state`, and `task_state`.

Normal response codes: 200

3.51.1.1. Request

This table shows the URI parameters for the show server extended status request:

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

This operation does not require a request body.

3.51.1.2. Response

Example 3.430. Show server extended status: JSON response

```
{
  "server": {
    "OS-EXT-STS:power_state": 1,
    "OS-EXT-STS:task_state": null,
    "OS-EXT-STS:vm_state": "active",
    "accessIPv4": "",
    "accessIPv6": "",
    "addresses": {
      "private": [
        {
          "addr": "192.168.0.3",
          "version": 4
        }
      ]
    },
    "created": "2013-02-07T19:35:09Z",
    "flavor": {
      "id": "1",
      "links": [
        {
          "href": "http://openstack.example.com/openstack/flavors/1",
          "rel": "bookmark"
        }
      ]
    },
    "hostId": "570eff4776ab310707d11d181037337197086998a8b3305c90bf87c8",
    "id": "ecb5e433-fa75-4db2-af3d-a29ae8618edc",
    "image": {
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",

```

```

        "links": [
          {
            "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b",
            "rel": "bookmark"
          }
        ],
      },
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/servers/ecb5e433-fa75-4db2-af3d-a29ae8618edc",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/servers/ecb5e433-fa75-4db2-af3d-a29ae8618edc",
          "rel": "bookmark"
        }
      ],
      "metadata": {
        "My Server Name": "Apache1"
      },
      "name": "new-server-test",
      "progress": 0,
      "status": "ACTIVE",
      "tenant_id": "openstack",
      "updated": "2013-02-07T19:35:10Z",
      "user_id": "fake"
    }
  }
}

```

Example 3.431. Show server extended status: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:OS-EXT-STS="http://docs.openstack.org/compute/ext/extended_status/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" status="ACTIVE" updated="2013-02-07T19:35:10Z" hostId="372afb648339fb6f22faa0b75fdd8834e2382fe02b352af8d7ee0b84" name="new-server-test" created="2013-02-07T19:35:09Z" userId="fake" tenantId="openstack" accessIPv4="" accessIPv6="" progress="0" id="68647408-85a7-4d9b-85e7-7f1e238983ad" OS-EXT-STS:vm_state="active" OS-EXT-STS:task_state="None" OS-EXT-STS:power_state="1">
  <image id="70a599e0-31e7-49b7-b260-868f441e862b">
    <atom:link href="http://openstack.example.com/openstack/images/70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  </image>
  <flavor id="1">
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel="bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="private">
      <ip version="4" addr="192.168.0.3"/>
    </network>
  </addresses>

```



```
<atom:link href="http://openstack.example.com/v2/openstack/servers/  
68647408-85a7-4d9b-85e7-7f1e238983ad" rel="self"/>  
<atom:link href="http://openstack.example.com/openstack/servers/  
68647408-85a7-4d9b-85e7-7f1e238983ad" rel="bookmark"/>  
</server>
```

3.51.2. List extended status for servers

Method	URI	Description
GET	/v2/{tenant_id}/servers/detail	Lists the extended status attributes in the detailed response for all servers.

The extended status attributes are `vm_state`, `power_state`, and `task_state`.

Normal response codes: 200

3.51.2.1. Request

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

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

This operation does not require a request body.

3.51.2.2. Response

Example 3.432. List extended status for servers: JSON response

```
{
  "servers": [
    {
      "OS-EXT-STS:power_state": 1,
      "OS-EXT-STS:task_state": null,
      "OS-EXT-STS:vm_state": "active",
      "accessIPv4": "",
      "accessIPv6": "",
      "addresses": {
        "private": [
          {
            "addr": "192.168.0.3",
            "version": 4
          }
        ]
      },
      "created": "2012-12-05T07:34:10Z",
      "flavor": {
        "id": "1",
        "links": [
          {
            "href": "http://openstack.example.com/openstack/
flavors/1",
            "rel": "bookmark"
          }
        ]
      },
      "hostId":
"585aa01f94eca692eff9f77ffe3eab866d8a819e97397e28c5c7df12",
      "id": "030758aa-5c41-41c6-8fb4-29d44eb96a85",
      "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
```

```

        "links": [
          {
            "href": "http://openstack.example.com/openstack/
images/70a599e0-31e7-49b7-b260-868f441e862b",
            "rel": "bookmark"
          }
        ],
      },
      "links": [
        {
          "href": "http://openstack.example.com/v2/openstack/
servers/030758aa-5c41-41c6-8fb4-29d44eb96a85",
          "rel": "self"
        },
        {
          "href": "http://openstack.example.com/openstack/servers/
030758aa-5c41-41c6-8fb4-29d44eb96a85",
          "rel": "bookmark"
        }
      ],
      "metadata": {
        "My Server Name": "Apache1"
      },
      "name": "new-server-test",
      "progress": 0,
      "status": "ACTIVE",
      "tenant_id": "openstack",
      "updated": "2012-12-05T07:34:10Z",
      "user_id": "fake"
    }
  ]
}

```

Example 3.433. List extended status for servers: XML response

```

<?xml version='1.0' encoding='UTF-8'?>
<servers xmlns:OS-EXT-STS="http://docs.openstack.org/compute/ext/
extended_status/api/v1.1" xmlns:atom="http://www.w3.org/2005/Atom" xmlns=
"http://docs.openstack.org/compute/api/v1.1">
  <server status="ACTIVE" updated="2012-12-05T07:35:57Z" hostId=
"20171312b8f2c42b69b09360e08d7fe257b2e021107be687d0302a96" name="new-
server-test" created="2012-12-05T07:35:56Z" userId="fake" tenantId=
"openstack" accessIPv4="" accessIPv6="" progress="0" id="085c76aa-a58f-45b8-
ba78-4d1e541d5f89" OS-EXT-STS:vm_state="active" OS-EXT-STS:task_state="None"
OS-EXT-STS:power_state="1">
  <image id="70a599e0-31e7-49b7-b260-868f441e862b">
    <atom:link href="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b" rel="bookmark"/>
  </image>
  <flavor id="1">
    <atom:link href="http://openstack.example.com/openstack/flavors/1" rel=
"bookmark"/>
  </flavor>
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <addresses>
    <network id="private">
      <ip version="4" addr="192.168.0.3"/>
    </network>
  </addresses>
</server>
</servers>

```

```

</addresses>
<atom:link href="http://openstack.example.com/v2/openstack/servers/
085c76aa-a58f-45b8-ba78-4d1e541d5f89" rel="self"/>
<atom:link href="http://openstack.example.com/openstack/servers/085c76aa-
a58f-45b8-ba78-4d1e541d5f89" rel="bookmark"/>
</server>
</servers>

```

3.52. Servers multiple create (servers)

Create one or more servers with an optional reservation ID. The request and response examples show how to create multiple servers with or without a reservation ID.

Method	URI	Description
POST	/v2/{tenant_id}/servers{? security_group,user_data, availability_zone, return_reservation_id,min_count, max_count}	Creates one or more servers with an optional reservation ID.

3.52.1. Create multiple servers

Method	URI	Description
POST	/v2/{tenant_id}/servers{? security_group,user_data, availability_zone, return_reservation_id,min_count, max_count}	Creates one or more servers with an optional reservation ID.

Normal response codes: 202

3.52.1.1. Request

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

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

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

Name	Type	Description
security_group	String <i>(Required)</i>	The name of the security group. If blank, the server is created in the default security group.
user_data	String <i>(Optional)</i>	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String <i>(Optional)</i>	The availability zone in which to launch the server.
return_reservation_id	String <i>(Optional)</i>	Set to <code>True</code> to generate a reservation ID for each server. Omit this attribute to create servers without a reservation ID. This extended attribute is enabled when the service provider enables multiple server launch.
min_count	String <i>(Optional)</i>	The minimum number of servers to launch when the service provider enables multiple server launch.
max_count	String <i>(Optional)</i>	The maximum number of servers to launch when the service provider enables multiple server launch.

Example 3.434. Create multiple servers: JSON request

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

```

```

        "path" : "/etc/banner.txt",
        "contents" :
"ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb2
="
    }
  ]
}
}

```

Example 3.435. Create multiple servers: XML request

```

<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  imageRef="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b"
  flavorRef="http://openstack.example.com/openstack/flavors/1"
  name="new-server-test"
  min_count="2"
  max_count="3">
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
dCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltcHVz
c2lvbi4uLnRoXMGaXMgdGhlIHBSYWNlIHRvIGdvIG5vdy4g
QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
ZSBwYXR0ZXJucyBiZWwhpbmQgYWxsIGNsb3VkcycwYmV5kIHlv
dSB3aWxsIGtub3csIHRvbywgd2hlbiB5b3UgbGlm dCB5b3Vy
c2VsZiBoaWdoIGVub3VnaCB0byBzZWUgYmV5b25kIGhvcml6
b25zLiINCgOKLVJpY2hhcmQgQmFjaA==
    </file>
  </personality>
</server>

```

Example 3.436. Create multiple servers: JSON request

```

{
  "server" : {
    "name" : "new-server-test",
    "imageRef" : "http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b",
    "flavorRef" : "http://openstack.example.com/openstack/flavors/1",
    "metadata" : {
      "My Server Name" : "Apache1"
    },
    "return_reservation_id": "True",
    "min_count": "2",
    "max_count": "3",
    "personality" : [
      {
        "path" : "/etc/banner.txt",
        "contents" :
"ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBpdCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb2
="
      }
    ]
  }
}

```

```
}

```

Example 3.437. Create multiple servers: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<server xmlns="http://docs.openstack.org/compute/api/v1.1"
  imageRef="http://openstack.example.com/openstack/images/
70a599e0-31e7-49b7-b260-868f441e862b"
  flavorRef="http://openstack.example.com/openstack/flavors/1"
  name="new-server-test"
  min_count="2"
  max_count="3"
  return_reservation_id="True">
  <metadata>
    <meta key="My Server Name">Apache1</meta>
  </metadata>
  <personality>
    <file path="/etc/banner.txt">
      ICAgICAgDQoiQSBjbG91ZCBkb2VzIG5vdCBrbm93IHdoeSBp
      dCBtb3ZlcyBpbjBqdXN0IHN1Y2ggYSBkaXJlY3Rpb24gYW5k
      IGF0IHN1Y2ggYSBzcGVlZC4uLk10IGZlZWxzIGFuIGltdGVs
      c2lvbi4uLnRoXMGaXMgdGhlIHBSYWNlIHRvIGdvIG5vdy4g
      QnV0IHRoZSBza3kga25vd3MgdGhlIHJlYXNvbnMgYW5kIHRo
      ZSBwYXR0ZXJucyBiZWhpbmQgYWxsIGNsb3VkcycwYmV5b25kIGhvcml6
      b25zLiINCgOKLVJpY2hhcmQgQmFjaA==
    </file>
  </personality>
</server>
```

This operation does not require a request body.

3.52.1.2. Response

Example 3.438. Create multiple servers: JSON response

```
{
  "server": {
    "adminPass": "wfksH3GTTseP",
    "id": "440cf918-3ee0-4143-b289-f63e1d2000e6",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/
440cf918-3ee0-4143-b289-f63e1d2000e6",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/
440cf918-3ee0-4143-b289-f63e1d2000e6",
        "rel": "bookmark"
      }
    ]
  }
}
```

Example 3.439. Create multiple servers: XML response

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

```
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" id="0857bea2-df83-4810-876a-093a6b9afb15" adminPass="edBzA3AnJP67">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/0857bea2-df83-4810-876a-093a6b9afb15" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/0857bea2-df83-4810-876a-093a6b9afb15" rel="bookmark"/>
</server>
```

Example 3.440. Create multiple servers: JSON response

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

Example 3.441. Create multiple servers: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.org/compute/api/v1.1" reservation_id="r-fq0lg4za"/>
```

This operation does not return a response body.

3.53. Servers with scheduler hints (servers)

Create a server with scheduler hints.

Method	URI	Description
POST	/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}	Creates a server with scheduler hints that are passed directly to the scheduler.

3.53.1. Create server with scheduler hints

Method	URI	Description
POST	/v2/{tenant_id}/servers{?security_group,user_data,availability_zone}	Creates a server with scheduler hints that are passed directly to the scheduler.

Normal response codes: 200

3.53.1.1. Request

This table shows the URI parameters for the create server with scheduler hints request:

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

This table shows the query parameters for the create server with scheduler hints request:

Name	Type	Description
security_group	String (Required)	The name of the security group. If blank, the server is created in the default security group.
user_data	String (Optional)	Configuration information or scripts to use upon launch. Must be Base64 encoded.
availability_zone	String (Optional)	The availability zone in which to launch the server.

Example 3.442. Create server with scheduler hints: JSON request

```
{
  "server": {
    "name": "new-server-test",
    "imageRef": "70a599e0-31e7-49b7-b260-868f441e862b",
    "flavorRef": "1"
  },
  "os:scheduler_hints": {
    "hypervisor": "xen",
    "near": "2b7c42eb-7736-4a0f-afab-f23969a35ada"
  }
}
```

Example 3.443. Create server with scheduler hints: XML request

```
<server
  xmlns="http://docs.openstack.org/compute/api/v1.1"
  xmlns:OS-SCH-HNT="http://docs.openstack.org/compute/ext/scheduler-
hints/api/v2"
  name='new-server-test'
  imageRef='70a599e0-31e7-49b7-b260-868f441e862b'
  flavorRef='1'
>
  <OS-SCH-HNT:scheduler_hints>
    <hypervisor>xen</hypervisor>
```

```
<near>eb999657-dd6b-464e-8713-95c532ac3b18</near>
</OS-SCH-HNT:scheduler_hints>
</server>
```

This operation does not require a request body.

3.53.1.2. Response

Example 3.444. Create server with scheduler hints: JSON response

```
{
  "server": {
    "adminPass": "yjzytFHb7XHc",
    "id": "f8f4f3ce-f6e0-4e05-8f79-bf984fdfce45",
    "links": [
      {
        "href": "http://openstack.example.com/v2/openstack/servers/
f8f4f3ce-f6e0-4e05-8f79-bf984fdfce45",
        "rel": "self"
      },
      {
        "href": "http://openstack.example.com/openstack/servers/
f8f4f3ce-f6e0-4e05-8f79-bf984fdfce45",
        "rel": "bookmark"
      }
    ]
  }
}
```

Example 3.445. Create server with scheduler hints: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns:atom="http://www.w3.org/2005/Atom" xmlns="http://docs.openstack.
org/compute/api/v1.1" id="a14a9bf5-93b3-49e4-85f1-01d7cf2645b4" adminPass=
"Q6jfHPrnvB9X">
  <metadata/>
  <atom:link href="http://openstack.example.com/v2/openstack/servers/
a14a9bf5-93b3-49e4-85f1-01d7cf2645b4" rel="self"/>
  <atom:link href="http://openstack.example.com/openstack/servers/
a14a9bf5-93b3-49e4-85f1-01d7cf2645b4" rel="bookmark"/>
</server>
```

This operation does not return a response body.